Intel® 41210 Serial to Parallel PCI Bridge

Developer's Manual

May 2005

Order Number: 278890-003US

Contents

Figures

Tables

Revision History

Date	Revision	Description
May 2005	003	Revised Table 1 and Table 9
October 2004	002	Updated PCI Express operation information in Section 1.1 and Table 2 inSection 2.2. Removed L0s state information throughout manual.
March 2004	001	Initial release

Introduction 1

The Intel® 41210 Serial to Parallel PCI Bridge (also called the 41210 Bridge or the 41210) integrates two PCI Express*-to-PCI/PCI-X bridges. Each bridge follows the PCI-to-PCI Bridge programming model. The PCI Express* port is compatible with the *PCI Express* Specification*, Revision 1.0a. The two PCI bus interfaces are comparable with the *PCI Local Bus Specification*, Revision 2.3 and the *PCI-X Addendum to the PCI Local Bus Specification*, Revision 1.0b.

1.1 PCI Express* Interface Features

- *PCI Express* Specification*, Revision 1.0a
- Support for single ×8, single ×4 or single x1 PCI Express* operation
- 64-bit addressing support
- 32-bit CRC (cyclic redundancy checking) covering all transmitted data packets
- 16-bit CRC on all link message information
- Raw bit-rate on the data pins of 2.5 Gbit/s, resulting in a raw bandwidth per pin of 250 MB/s
- Maximum realized bandwidth on PCI Express* interface of 2 GB/s (in ×8 mode) in each direction simultaneously, for an aggregate of 4 GB/s

1.2 PCI-X Interface Features

- *PCI Local Bus Specification*, Revision 2.3
- *PCI-to-PCI Bridge Specification*, Revision 1.1
- *PCI-X Addendum to the PCI Local Bus Specification*, Revision 1.0b
- 64-bit 66 MHz, 3.3 V, ***not*** 5 V tolerant
- On-Die Termination (ODT) with 8.3 KΩ pull-up to 3.3 V for PCI signals
- Six external REQ/GNT pairs for internal arbiter on segment A and B respectively
- Programmable bus parking on either the last agent or always on Intel® 41210 Serial to Parallel PCI Bridge
- Two-level programmable round-robin internal arbiter with Multi-Transaction Timer (MTT)
- External PCI clock-feed support for asynchronous primary and secondary domain operation
- 64-bit addressing for upstream and downstream transactions
- Downstream LOCK# support
- No upstream LOCK# support
- PCI fast Back-to-Back capable as target
- Up to four active and four pending upstream memory read transactions

- Up to two downstream delayed (memory read, I/O read/write and configuration read/write) transactions
- Tunable inbound read prefetch algorithm for PCI MRM/MRL commands
- Device hiding support for secondary PCI devices
- Secondary bus private memory support via opaque memory region
- Local initialization via SMBus
- Secondary side initialization via Type 0 configuration cycles
- Full peer-to-peer read/write capability between the two secondary PCI segments

1.3 Power Management

- Support for PCI PM 1.1-compatible D0, D3hot and D3cold device power states
- Support for PME# event propagation on behalf of PCI devices

1.4 SMBus Interface

- Compatible with *System Management Bus Specification*, Revision 2.0
- Slave-mode operation only
- Full read/write access to all configuration registers

1.5 JTAG

- *IEEE Standard Test Access Port and Boundary Scan Architecture 1149.1a*

Signal Description 2

The "#" symbol at the end of a signal name indicates that the active (asserted) state occurs when the signal is at a low voltage level. When "#" is not present after the signal name, the signal is asserted when at the high voltage level. The following notations are used to describe the signal type:

I: Input pin

O: Output pin

OD: Open-drain Output pin

I/O: Bidirectional Input/Output pin

I/OD: Bidirectional Input/Open-drain Output pin

2.1 On-Die Termination (ODT)

The Intel® 41210 Serial to Parallel PCI Bridge (also called the 41210 Bridge or the 41210) incorporates On-Die Termination (ODT) for most of the PCI interface signals. ODT eliminates the need for the system designer to incorporate external pull-up resistors in the design.

Table 1, "ODT Signals" on page 14 lists the signals that have an on-die termination of 8.33 KΩ @ 40%.

Table 1. ODT Signals

A_ACK64#	B_ACK64#
A_AD[63:32]	B_AD[63:32]
A_CBE#[7:4]	B_CBE#[7:4]
A_DEVSEL#	B_DEVSEL#
A_FRAME#	B_FRAME#
A_GNT#[5:0]	B_GNT#[5:0]
A_IRDY#	B_IRDY#
A_PAR	B_PAR
A_PAR64	B_PAR64
A_PERR#	B_PERR#
A_LOCK#	B_LOCK#
A_REQ#[5:0]	B_REQ#[5:0]
A_REQ64#	B_REQ64#
A_SERR#	B_SERR#
A_STOP#	B_STOP#
A_TRDY#	B_TRDY#
A_INTA#	B_INTA#
A_INTB#	B_INTB#
A_INTC#	B_INTC#
A_INTD#	B_INTD#
TCK	
TDI	
TDO	
TMS	

2.2 PCI Express* Interface

Table 2. PCI Express* Interface Pins

Signal	I/O	Description
REFCLKp/ REFCLKn	I	PCI Express* **Reference Clocks:** 100 MHz differential clock pair
PETp[7:0]/ PETn[7:0]	O	PCI Express* **Serial Data Transmit:** PCI Express* differential data transmit signals X8 Mode: All PETp[7:0]/PETn[7:0] are used. X4 Mode: Only PETp[3:0]/PETn[3:0] are used. X1 Mode: Either PETp[0]/PETn[0] is used or PETp[7]/PETn[7] is used.
PERp[7:0]/ PERn[7:0]	I	PCI Express* **Serial Data Receive:** PCI Express* differential data receive signals X8 Mode: All PERp[7:0]/PERn[7:0] are used. X4 Mode: Only PERp[3:0]/PERn[3:0] are used. X1 Mode: Either PERp[0]/PERn[0] is used or PERp[7]/PERn[7] is used.
PE_RCOMP[1:0]	I	PCI Express* **Compensation Inputs:** Analog signals. Connect to a 24.9 Ω ±1% pull-up resistor to 1.5 V. A single resistor can be used for both signals.
Total	36	

2.3 PCI Bus Interface (Two Instances)

Each interface is marked by either the letter “A” or “B” to signify the interface. For example, A_AD refers to the AD bus on PCI bus A, and B_AD refers to the AD bus on PCI bus B. For pin names described in the following sections, an “X” in the name indicates either A or B, for the PCI bus A and PCI bus B sides, respectively. For example, “X_PAR” indicates A_PAR on the PCI bus A and B_PAR on the PCI bus B.

Table 3. PCI Interface Pins (Sheet 1 of 2)

Signal	I/O	Description
A_AD[31:0] B_AD[31:0]	I/O	**PCI Address/Data:** These signals are a multiplexed address and data bus. During the address phase or phases of a transaction, the initiator drives a physical address on X_AD[31:0]. During the data phases of a transaction, the initiator drives write data, or the target drives read data. No external pull-up resistors are required on the system board for these signals.
A_C/BE#[3:0] B_C/BE#[3:0]	I/O	**Bus Command and Byte Enables:** These signals are a multiplexed command field and byte enable field. During the address phase or phases of a transaction, the initiator drives the transaction type on C/BE#[3:0]. When there are two address phases, the first address phase carries the dual address command and the second address phase carries the transaction type. For both read and write transactions, the initiator drives byte enables on C/BE#[3:0] during the data phases. No external pull-up resistors are required on the system board for these signals.
A_PAR B_PAR	I/O	**Parity:** Even parity is calculated on 36 bits—AD[31:0] plus C/BE[3:0]#. It is calculated on all 36 bits regardless of the valid byte enables. It is generated for address and data phases. It is driven identically to the AD[31:0] lines, except it is delayed by exactly one PCI clock. It is an output in the following cases: • During the address phase for all transactions initiated by the Intel® 41210 Serial to Parallel PCI Bridge • During all data phases when the 41210 is the initiator of a PCI write transaction • When the 41210 is the target of a read transaction The 41210 checks parity when it is the initiator of PCI read transactions and when it is the target of PCI write transactions. No external pull-up resistors are required on the system board for these signals.
A_DEVSEL# B_DEVSEL#	I/O	**Device Select:** The bridge asserts DEVSEL# to claim a PCI transaction. As a target, the 41210 asserts DEVSEL# when a PCI master peripheral attempts to access an address destined for PCI Express*. As an initiator, DEVSEL# indicates the response to a transaction initiated by the 41210 on the PCI bus. DEVSEL# is tristated from the leading edge of PCIRST#. DEVSEL# remains tristated by the 41210 until driven as a target. No external pull-up resistors are required on the system board for these signals.
A_FRAME# B_FRAME#	I/O	**Frame:** FRAME# is driven by the initiator to indicate the beginning and duration of an access. While FRAME# is asserted, data transfers continue. When FRAME# is deasserted the transaction is in the final data phase. No external pull-up resistors are required on the system board for these signals.
A_IRDY# B_IRDY#	I/O	**Initiator Ready:** IRDY# indicates the ability of the initiator to complete the current data phase of the transaction. A data phase is completed when both IRDY# and TRDY# are sampled asserted. No external pull-up resistors are required on the system board for these signals.
A_TRDY# B_TRDY#	I/O	**Target Ready:** TRDY# indicates the ability of the target to complete the current data phase of the transaction. A data phase is completed when both TRDY# and IRDY# are sampled asserted. TRDY# is tristated from the leading edge of RST#. TRDY# remains tristated by the 41210 until driven as a target. No external pull-up resistors are required on the system board for these signals.
A_STOP# B_STOP#	I/O	**Stop:** This bit indicates that the target is requesting an initiator to stop the current transaction. No external pull-up resistors are required on the system board for these signals.

Table 3. PCI Interface Pins (Sheet 2 of 2)

Signal	I/O	Description
A_PERR# B_PERR#	I/O	**Parity Error:** PERR# is driven by an external PCI device when it receives data that has a parity error. PERR# is driven by the 41210 in the following cases: • when the 41210, as an initiator, detects a parity error during a read transaction • when the 41210, as a target, detects a parity error during a write transaction No external pull-up resistors are required on the system board for these signals.
A_SERR# B_SERR#	I	**System Error:** The 41210 samples SERR# as an input and conditionally forwards it to the PCI Express*. No external pull-up resistors are required on the system board for these signals.
A_REQ#[5:0] B_REQ#[5:0]	I	**PCI Requests:** REQ# receives request inputs into the internal arbiter. No external pull-up resistors are required on the system board for these signals.
A_GNT#[5:0] B_GNT#[5:0]	O	**PCI Grants:** GNT# is the bus grant output corresponding to request input bits[5:0] from the internal arbiter. GNT# indicates that an initiator can start a transaction on the PCI bus. No external pull-up resistors are required on the system board for these signals.
A_M66EN B_M66EN	I/OD	**66 MHz Enable:** M66EN is an input signal from the PCI bus that indicates the speed of the PCI bus. When it is high, the bus speed is 66 MHz. When it is low, the bus speed is 33 MHz. This signal is used to generate an appropriate clock (33 or 66 MHz) on the PCI bus. To tie high: Use an approximately 8.2 KΩ resistor to pull to VCC33. To tie low: Pull down to ground.
A_PCIXCAP B_PCIXCAP	I	**PCI-X Capable:** PCIXCAP indicates whether all devices on the PCI bus are PCI-X devices, so that the 41210 can switch into PCI-X mode. Use an approximately 8.2 KΩ resistor to pull to VCC33.
A_LOCK# B_LOCK#	O	**PCI Lock:** LOCK# indicates an exclusive bus operation and may require multiple transactions to complete. This signal is an output from the bridge when it is initiating exclusive transactions on PCI. LOCK# is ignored when PCI masters are granted the bus. Locked transaction do not propagate upstream. No external pull-up resistors are required on the system board for these signals.
Total	118	

2.4 PCI Bus Interface 64-Bit Extension (Two Interfaces)

Table 4. PCI Interface Pins: 64-Bit Extensions

Signal	I/O	Description
A_AD[63:32] B_AD[63:32]	I/O	**PCI Address/Data:** The AD signals are a multiplexed address and data bus. This bus provides an additional 32 bits to the PCI bus. During the data phases of a transaction, the initiator drives the upper 32 bits of 64-bit write data, or the target drives the upper 32 bits of 64-bit read data, when REQ64# and ACK64# are both asserted.
A_C/BE#[7:4] B_C/BE#[7:4]	I/O	**Bus Command and Byte enables upper 4 bits:** The C/BE# signals are a multiplexed command field and byte enable field. For both reads and write transactions, the initiator drives byte enables for the AD[63:32] data bits on C/BE[7:4] during the data phases when REQ64# and ACK64# are both asserted.
A_PAR64 B_PAR64	I/O	**PCI interface upper 32 bits parity:** PAR64 carries the even parity of the 36 bits of AD[63:32] and C/BE#[7:4] for both address and data phases.
A_REQ64# B_REQ64#	I/O	**PCI interface request 64-bit transfer:** REQ64# is asserted by the initiator to indicate that the initiator is requesting a 64-bit data transfer. REQ64# has the same timing as FRAME#. When the 41210 is the initiator, this signal is an output. When the 41210 is the target, this signal is an input.
A_ACK64# B_ACK64#	I/O	**PCI interface acknowledge 64-bit transfer:** ACK64# is asserted by the target only when REQ64# is asserted by the initiator, to indicate the target ability to transfer data using 64 bits. ACK64# has the same timing as DEVSEL#.
Total	78	

2.5 PCI Bus Interface Clocks and, Reset and Power Management (Two Interfaces)

Table 5. PCI Clock and Reset Pins

Signal	I/O	Description
A_CLKO[6:0] B_CLKO[6:0]	O	**PCI Clock Output:** CLKO is the 33/66/100/133 MHz clock for a PCI device. X_CLK[6] must be connected to the respective X_CLKIN input for feeding the PCI interface logic. Unused clock outputs may be disabled via the "Offset 43h: PCLKC—PCI Clock Control" register and should be treated as no connects on the board.
A_CLKIN B_CLKIN	I	**PCI Clock In:** CLKIN is the PCI clock feedback input. CLKIN must be connected to the corresponding X_CLKO[6] through a 22 Ω ± 1% series resistor.
A_RST# B_RST#	O	**PCI Reset:** The bridge asserts RST# to reset devices that reside on the secondary PCI bus.
A_PME# B_PME#	I	**PCI Power Management Event:** PME# is the PCI bus power management event signal. PME# is a shared open-drain input from all the PCI cards on the corresponding PCI bus segment. PME# is a level-sensitive signal that is converted to a PME event on PCI Express*. PME# does not have on-die 8.3 KΩ pull-up. This pull-up must be provided externally.
Total	20	

2.6 Interrupt Interface (Two Interfaces)

This section lists the interrupt interface signals. There are two sets of interrupt signals for the standard INTA–INTD PCI signals.

Table 6. Interrupt Interface Pins

Signal	I/O	Description
A_INTA# A_INTB# A_INTC# A_INTD# B_INTA# B_INTB# B_INTC# B_INTD#	I	**Interrupt Request Bus:** The interrupt lines from PCI interrupts INTA#–INTD# can be routed to these interrupt lines. Routing must be based on device number in accordance with the instructions given in Section 7, “Interrupt Support” on page 53.
Total	8	

2.7 Reset Straps

The following signals are used for static configuration. These signals are all sampled on the rising edge of PERST#.

Table 7. Reset Strap Pins

<table>
<tr><th>Signal</th><th>I/O</th><th>Description</th></tr>
<tr><td>A_133EN
B_133EN</td><td>I</td><td>PCI-X 133 MHz Enable: The 133EN pin, when high, allows the PCI-X segment to run at 133 MHz when X_PCIXCAP is sampled high. When 133EN is low, the PCI-X segment runs only at 100 MHz when X_PCIXCAP is sampled high.
To tie high: Use an approximately 8.2 KΩ resistor to pull to VCC33.
To tie low: Pull down to ground.</td></tr>
<tr><td>A_STRAP[6:0]
B_STRAP[6:0]</td><td>I</td><td>Internal Test Modes: For normal operation, X_STRAP[6] and [2:0] must be pulled low and X_STRAP[5:3] must be pulled high, as shown in the table below.
<table>
<tr><th>X_STRAP</th><th>Logic Level</th></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>2</td><td>0</td></tr>
<tr><td>3</td><td>1</td></tr>
<tr><td>4</td><td>1</td></tr>
<tr><td>5</td><td>1</td></tr>
<tr><td>6</td><td>0</td></tr>
</table>
To tie high: Use approximately an 8.2 KΩ resistor to pull up to VCC33.
To tie low: Pull down to VSS.</td></tr>
<tr><td>A_TEST[2:1]
B_TEST{2:1]</td><td>I</td><td>Internal Test Modes: These straps must be pulled high to VCC33. Use an approximately 8.2 KΩ resistor to pull up to VCC33.</td></tr>
<tr><td>CFGRETRY</td><td>I</td><td>Configuration Retry: This pin, when sampled high, sets the Configuration Cycle Retry Bit (bit 3) in the Bridge Initialization Register (“Offset FCh: BINIT—Bridge Initialization Register” on page 104).
When no local initialization is needed, this pin must be pulled low to VSS.
See Section 9, “Local Initialization” for additional details.</td></tr>
<tr><td>Total</td><td>19</td><td></td></tr>
</table>

2.8 SMBus Interface

Table 8. SMBus Interface Pins

Signal	I/O	Description
SMBCLK	I/OD	**SMBus Clock:** This signal must be pulled to 3.3 V through an 8.2 KΩ resistor.
SMBDAT	I/OD	**SMBus Data:** This signal must be pulled to 3.3 V through an 8.2 KΩ resistor.
SMBUS[5] SMBUS[3:1]	I	**SMBus Addressing Straps:** These straps set the SMBus address for the 41210 Bridge. The address is determined as indicated below: • Bit[7] 1 • Bit[6] 1 • Bit[5] SMBUS[5] • Bit[4] 0 • Bit[3] SMBUS[3] • Bit[2] SMBUS[2] • Bit[1] SMBUS[1] These signals (bits[5], [3:1]) must be pulled up to 3.3 V or down to ground. Sampled at the rising edge of PERST#.
Total	6	

2.9 Miscellaneous Pins

Table 9. Miscellaneous Pins

Signal	I/O	Description
CFGRST#	O	**Configuration Reset:** This signal is asserted low when ever the bridge goes through a fundemental reset (PERST#, RSTIN#, or PCI Express Reset). This signal should be used to indicate when the local initialization methods should be executed. Refer to the *Intel® 41210 Serial to Parallel PCI Bridge Design Guide* for more information.
PERST#	I	**PCI Express Fundamental Reset:** When low, asynchronously resets the internal logic (including sticky bits).
RSTIN#	I	**Reset In:** When Asserted, this signal asynchronously resets the internal logic and asserts X_RST# output for both PCI interfaces. This signal should be pulled high for adapter card usage.
TCK	I	**TAP Clock In**: This is the input clock to the JTAG TAP controller. Acceptable frequency is 0-16MHz If not utilizing JTAG, this signal can be left as a no connect.
TDI	I	**Test Data In:** This is the serial data input to the JTAG BSCAN shift register chain and to the JTAG BSCAN control logic. This is latched in on the rising edge of TCK. If not utilizing JTAG, this signal can be left as a no connect.
TDO	O	**Test Data Output:** This is the serial data output from the JTAG BSCAN logic If not utilizing JTAG, this signal can be left as a no connect.
TMS	I	**Test Mode Select:** This signal controls the TAP controller state machine to move to different states and is sampled on the rising edge of TCK. If not utilizing JTAG, this signal can be left as a no connect.
TRST#	I	**Test Reset In:** This signal is used to asynchronously reset the JTAG BSCAN logic. If not utilizing JTAG, connect this signal to ground through a 1KΩ pull-down resistor.
RESERVED[8:1]	I	**Reserved:** (8 pins) These input pins should be pulled low Use an approximately 8.2KΩ resistor to pull-down to ground.
NC[19:18], NC[16:1] A_NC[10:1] B_NC[10:1]	O	**No Connect:** (39 pins) These output pins should be left floating
NC[17]	O	This signal requires an external pull-up, 8.2K ohm to 3.3V
Total	57	

2.10 Voltage Pins

Table 10. Miscellaneous Pins

Signal	Number	Description
RCOMP	1	**Analog Compensation Pin:** RCOMP is the analog compensation pin for PCI. Pull down to ground through a 100 Ω ±1% resistor.
VCC	36	**1.5 V Core Voltage:** 1.5 V ± 5%.
VCCAPE	1	**Analog** PCI Express* **Voltage:** • DC: 1.5 V ± 3% • AC: ±5 mV above 1 MHz at package pin under DC load conditions
VCCAPCI[2:0]	3	**Analog PCI Voltages:** 1.5 V ±3%
VCCBGPE	1	**Analog Band-gap Voltage:** • DC: 2.5 V ±3% • AC: ±10 mV above 1 MHz at package pin under DC load conditions
VSSBGPE	1	**Analog Band-gap Ground**
VCCPE	9	**1.5 V** PCI Express* **Voltage:** 1.5 V ±3%
VCC33	30	**3.3 V PCI I/O Voltage:** 3.3 V ±5%.
VSS	142	**Ground:** Ground for all voltage rails
VSSAPE	1	**Analog** PCI Express* **Ground**
Total	225	

§ §

THIS PAGE INTENTIONALLY LEFT BLANK

PCI-X Interface 3

This section deals with the specifics of the operation and transaction flow details of the PCI interfaces.

3.1 Initialization

The Intel® 41210 Serial to Parallel PCI Bridge (also called the 41210 Bridge or the 41210) is the source bridge for the PCI bus and senses the X_M66EN, X_133EN, and X_PCIXCAP pins to decide the mode and frequency of operation. Encoding on the M66EN and PCIXCAP pins along with the PCI reset straps is shown below; these encodings identify the capabilities of the system.

Table 11. PCI Mode Pin/Strap Encoding

X_133EN[1]	X_M66EN	X_PCIXCAP	PCI Bus Mode	PCI Frequency
–	Ground	Ground	PCI conventional	33 MHz
–	Not connected	Ground	PCI conventional	66 MHz
–	–	Pull-down	PCI-X	66 MHz
Ground	–	Not connected	PCI-X	100 MHz
Not connected	–	Not connected	PCI-X	133 MHz

NOTE:
1. This pin is used by the board designer to run the PCI-X bus at 100 MHz even when all the PCI-X devices on the bus are 133 MHz capable, so as to accommodate the board routing limitation on frequency. Note that to accommodate running a PCI-X bus at 66 MHz, when all cards are capable of 133 MHz, the motherboard has to drive the PCIXCAP pin to VCC33/2.

As soon as the 41210 identifies the capabilities of the PCI bus devices, it drives the initialization pattern on the DEVSEL#, STOP#, TRDY#, FRAME#, and IRDY# pins as per Table 12 to initialize the PCI bus devices to the proper mode and frequency. The patterns shown in Table 12 below are stable on the rising edge of the **A_RST#** and **B_RST#** pins. Refer to the *PCI-X Addendum to the PCI Local Bus Specification*, Revision 1.0b for details on the timing of these patterns with respect to the **A_RST#** and **B_RST#** pins.

Table 12. PCI-X Initialization Pattern

FRAME# and IRDY# are Deasserted							
DEVSEL#	STOP#	TRDY#	Mode	Clock Period (ns)		Clock Freq. (MHz)	
				Minimum	Maximum	Minimum	Maximum
Deasserted	Deasserted	Deasserted	PCI 33	30	—	0	33
			PCI 66	15	30	33	66
Deasserted	Deasserted	Asserted	PCI-X	15	20	50	66
Deasserted	Asserted	Deasserted	PCI-X	10	15	66	100
Deasserted	Asserted	Asserted	PCI-X	7.5	10	100	133
Asserted	Deasserted	Deasserted	PCI-X	Reserved			
Asserted	Deasserted	Asserted	PCI-X				
Asserted	Asserted	Deasserted	PCI-X				
Asserted	Asserted	Asserted	PCI-X				

In summary:

- A_RST# and B_RST# are outputs from the 41210.
- PCI clocks are actively driven out from the 41210.
- The 41210 drives X_AD[31:0], X_BE[3:0], and X_PAR low during PCI bus reset.
- The 41210 drives X_REQ64# low during reset.

3.2 Transactions Supported

3.2.1 PCI Mode

Table 13 lists all the transactions supported by the 41210 on the PCI bus. The 41210 supports full 64-bit addressing upstream and downstream and can both generate and accept dual address cycles.

Table 13. PCI Transactions Supported

Transaction	Encoding[1]	Master	Target
Interrupt acknowledge	0000	No	No
Special cycle (PCI Express* Type1-to-PCI Special Cycle)	0001	Yes	No
I/O read	0010	Yes	Yes[2]
I/O write	0011	Yes	Yes
Memory read	0110	Yes	Yes
Memory write	0111	Yes	Yes
Configuration read	1010	Yes	Yes[3]
Configuration write	1011	Yes	Yes
Memory read multiple	1100	No	Yes
Dual address cycle	1101	Yes	Yes
Memory read line	1110	No	Yes
Memory write and invalidate	1111	No	Yes
LOCK transaction	–	Yes	No

NOTES:
1. PCI command encodings that are not shown in this table are ignored.
2. I/O transactions are forwarded from PCI to PCI Express* only when the inbound I/O enable bit is set in the "Offset FCh: BINIT—Bridge Initialization Register" on page 104
3. Upstream Type 0 configuration cycles to the 41210 Bridge's own configuration space are supported.

3.2.2 PCI-X Mode

Table 14 lists the transactions that the 41210 supports when the PCI interface is in the PCI-X mode. As a master, the 41210 supports the memory write block command for writes that are multiples of cache-line.

Table 14. PCI-X Transactions Supported

Transaction	Encoding[1]	Master	Target
Interrupt acknowledge	0000	No	No
Special cycle (PCI Express* Type1-to-PCI Special Cycle)	0001	Yes	No
I/O read	0010	Yes	Yes
I/O write	0011	Yes	Yes
Reserved	0100	No	No
Reserved	0101	No	No
Memory read DWORD	0110	Yes	Yes
Memory write	0111	Yes	Yes
Alias to memory read block	1000	No	Yes
Alias to memory write block	1001	No	Yes
Configuration read	1010	Yes	Yes[2]
Configuration write	1011	Yes	Yes
Split completion	1100	Yes	Yes
Dual address cycle	1101	Yes	Yes
Memory read block	1110	Yes	Yes
Memory write block	1111	Yes	Yes
LOCK transaction	–	Yes	No

NOTES:
1. PCI command encodings that are not detailed in this table are ignored.
2. Upstream Type 0 configuration cycles to the bridge's own configuration space are supported.

3.2.3 Read Transactions

3.2.3.1 Prefetchable

Any memory read line or memory read multiple commands on PCI that are decoded by the 41210 are prefetched on the PCI Express* interface. The prefetchability of a given PCI read request is determined by the prefetch policy (PP) bits[55:54] of the "Offset 178h: PREFCTRL—Prefetch Control Register" on page 119. The amount of data prefetched depends on the clock frequency, x_REQ64#, and the command type. The 41210 does not prefetch past a 4 KB page boundary.

3.2.3.2 Delayed

All memory read transactions are delayed read transactions. When the 41210 accepts a delayed read request, it samples the address, command, and address parity. This information is entered into the delayed transaction queue. When the 41210 is in PCI-X mode, transactions follow the split transaction model of PCI-X. Read data returned from PCI Express* for an active delayed transaction entry is forwarded to the PCI-X master as a split completion.

3.2.3.3 Inbound Read Request Algorithm

In PCI mode:

- Each read stream always gets exactly 1 K buffer—no more or no less.
- A maximum of four requests can be outstanding per stream/buffer.
- A maximum of eight requests can be outstanding per PCI segment.

In PCI-X mode:

- Each read stream requests and gets buffer in 512 B chunks.
- There is no limit on how many 512 B chunks a read stream can occupy.
- A maximum of four requests can be outstanding per stream.
- A maximum of one request can be outstanding per 512 B buffer.
- A maximum of eight requests can be outstanding per PCI-X segment.

3.2.4 Configuration Transactions

Type 0 configuration transactions are issued when the intended target resides on the same PCI bus as the initiator. A Type 0 configuration transaction is identified by the configuration command and by the lowest two bits of the address being set to 00b.

Type 1 configuration transactions are issued when the intended target resides on another PCI bus, or when a special cycle is to be generated on another PCI bus. A Type 1 configuration transaction is identified by the configuration command and by the lowest two bits of the address being set to 01b.

The register number is found in both Type 0 and Type 1 formats and gives the Dword address of the configuration register to be accessed. The function number is also included in both Type 0 and Type 1 formats and indicates which function of a multifunction device is to be accessed. For single-function devices, this value is not decoded. Type 1 configuration transaction addresses also include a five-bit field designating the device number that identifies the device on the target PCI bus that is to be accessed. In addition, the bus number in Type 1 transactions specifies the PCI bus to which the transaction is targeted.

3.2.5 LOCK Cycles

A lock is established when **all** the following conditions are true:

- A PCI Express* device initiates a Memory Read Lock (MRdLk) request to read from a target PCI device.
- LOCK# is asserted on the PCI bus.
- The target PCI device responds with a **TRDY#**.

The bus is unlocked when the Unlock Message is received on PCI Express*.

When the PCI bus is locked, all upstream memory transactions from that bus are retried. The 41210 upstream read prefetch engine stops issuing any more requests on the PCI Express* bus. However, note that the 41210 accepts read completions for upstream read requests that were issued before the lock was established on the PCI bus when they return on PCI Express*.

As soon as the bus is locked, any PCI Express* cycle to PCI is driven with the **LOCK#** pin asserted, even when that particular cycle is not locked. This is not expected to occur, because under lock, peer-to-peer accesses are internally blocked and the upstream component must not send any non-locked transactions downstream.

When one PCI bus segment is locked, the other is still free to accept cycles (in other words, that bus is not locked. However, these transactions are not allowed to proceed on PCI Express* or the locked PCI segment). Therefore, as soon as the PCI bus is locked, additional cycles do not proceed onto PCI Express* from the non-locked PCI segment.

During the LOCK sequence, when the initial locked read command results in a master or target abort (either on the PCI bus or the internal switch interconnect), the 41210 does not establish lock, and it sends a completion packet on PCI Express* with an error status. In case of a subsequent memory read or memory write receiving a target or master abort during a LOCK sequence, the 41210 unlocks only after the unlock message is received on PCI Express*.

- Downstream LOCK is supported by the 41210.
- Upstream LOCK transactions are treated with the LOCK signal ignored.

See Table 15 below for a summary of the 41210 responses to LOCK transactions.

Table 15. LOCK Transaction Handling in the Intel® 41210 Serial to Parallel PCI Bridge

End Point	Source	
	PCI	PCI Express*
PCI	–	Forward to PCI w/ LOCK#
PCI Express*	Ignore[1]	–

NOTE:
1. Transaction is treated as if it is a normal read or write transaction.

3.2.6 Decoding

In the PCI mode, the 41210 supports only the linear increment address mode for bursting memory transfers (indicated when the lowest two address bits are equal to 0). When either of these address bits is non-zero, the 41210 disconnects the transaction after the first data transfer. The 41210 decodes all PCI cycles with medium DEVSEL# timing. In the PCI-X mode, 41210 always decodes as a Type A target. Also, in PCI-X mode, the 41210 decodes split completions using the primary bus number field.

Refer to Section 5, “Addressing” on page 41 for a general description of addressing and decoding.

3.2.7 Transaction Termination

3.2.7.1 PCI Mode Transaction Termination

- Normal Termination

  As a PCI master, the 41210 uses normal termination when DEVSEL# is returned by the target within five clock cycles of FRAME# assertion. It terminates a transaction when one of the following conditions are met:

  — All write data for a write transaction are transferred from the 41210 data buffers to the target (the 41210 does not generate fast back-to-back transactions).

  — All read data for a read transaction are transferred from the target to the 41210.

  — The master latency timer expires and the bus grant of the 41210 is de-asserted.

- Master Abort

  When the transaction initiated by the 41210 does not receive a DEVSEL# response within five clocks of FRAME# assertion, the 41210 terminates the transaction with a master abort. The 41210 sets the received master abort bit in the secondary status register. Read requests (configuration, I/O, memory) that receive master abort termination are sent back to PCI Express*/peer PCI with a master abort status.

  Note that when the 41210 performs a Type 1 to special cycle translation, a master abort is the expected termination for the special cycle on the target bus. In this case, the master abort received bit is not set, and the Type 1 configuration transaction is disconnected after the first data phase.

- Target Abort

  When the 41210 receives a target abort, and the cycle requires completion on PCI Express*, the bridge returns the target abort status to PCI Express*. The 41210 sets the received target abort status bit in the secondary status register for all target aborts it receives on the PCI bus.

  Target abort can occur during any data phase of a PCI-X transaction. A read completion packet to PCI Express*/peer PCI, incurring a target abort in the middle of the packet, returns valid data to the point of target abort and a target abort completion status for the reminder.

- Disconnect and Retry

  When the 41210 receives a disconnect response from a target, it re-initiates the transfer with the remaining length. When the 41210 receives a retry, it waits at least two PCI clocks before it retries the transaction. When the retried transaction is a write, the 41210 retries the write until it completes normally or with a target or master abort. When the retried transaction is a delayed read or delayed write transaction, the 41210 allows memory reads and writes to pass the transaction. Refer to Section 6, “Transaction Ordering” on page 51 for details on the kinds

of reordering allowed. Retry is not considered an error condition, so there is no error logging or reporting done on a retry.

- The 41210 terminates a transaction with retry to an initiator when one of the following conditions is met:
  - The 41210 receives a new memory read transaction, and the 41210 delayed transaction queue is full.
  - The 41210 receives a memory read that has already been queued, but has not completed on PCI Express*.
  - The 41210 receives a memory read that has been queued and completed on PCI Express*, but ordering rules require a downstream posted write to complete ahead of it.
  - A LOCK transaction has been established from PCI Express*-to-PCI.
  - The 41210 receives a memory write transaction, and the 41210 has no free buffer space to accept the write.
  - A memory write transaction is from a master other than the master that was previously retried (starvation prevention mechanism).
- The 41210 disconnects an initiator when one of the following conditions is met:
  - The 41210 cannot accept any more write data.
  - The 41210 has no more read data to deliver.
  - The memory address is non-linear.
  - The inverse decode window ends.
- The 41210 returns a target abort to PCI when the following condition is met.
  - The cycle master aborted, or the target aborted on PCI Express*/peer PCI.

3.2.7.2 PCI-X Mode Transaction Termination

- Initiator Disconnect or Satisfaction of Byte Count

  As a PCI-X master, the 41210 uses normal termination (initiator disconnect or satisfaction of byte count) if DEVSEL# is returned by the target within six clock cycles after address phase. The 41210 terminates a transaction when one of the following conditions are met:

  — Initiator disconnect occurs when all write data indicated in the byte count of the write transaction is transferred from the 41210 data buffers to the target. The 41210 does not perform an initiator disconnect on a write before the byte count size has been satisfied.

  — Initiator disconnect at the next ADB on a split read completion because the 41210 data buffer has run dry.

  — Initiator disconnect occurs at the next ADB when the master latency timer expires and the bus grant of the 41210 is de-asserted.

- Master Abort Termination

  When a transaction initiated by the 41210 does not receive a DEVSEL# response within six clocks after address phase, the 41210 terminates the transaction with a master abort. The 41210 sets the received master abort bit in the secondary status register. Read requests (configuration, I/O, memory) that receive master abort termination are sent back to PCI Express*/peer PCI with a master abort status. Delayed write requests that receive master abort are sent back to PCI Express* with a master abort status.

Note: When the 41210 performs a Type 1 to special cycle translation, a master abort is the expected termination for the special cycle on the target bus. In this case, the master abort received bit is not set, and the Type 1 configuration transaction is disconnected after the first data phase.

- Target Abort

  When the 41210 receives a target abort, and the cycle requires completion on PCI Express*, the 41210 returns the target abort status to PCI Express*. The 41210 sets the received target abort status bit in the secondary status register for all target aborts it receives on the PCI bus. Target abort can happen on any data phase of a PCI-X transaction. A read completion packet to PCI Express*/peer PCI that incurs a target abort in the middle of the packet returns valid data to the point of target abort, all 1s for the reminder of the length, and a target abort completion status for the entire packet.

- Disconnect and Retry

  When the 41210 receives a disconnect response (single data phase or at next ADB) from a target, it re-initiates the transfer with the remaining length. When the 41210 receives a retry, it waits at least two PCI clocks before it retries the transaction. When the retried transaction is a write, the 41210 retries the write until it completes normally or with a target or master abort. When the retried transaction is a delayed read or delayed write transaction, the 41210 allows memory reads, split completions, and writes to pass the transaction. Refer to Section 6, “Transaction Ordering” on page 51 for details on the kinds of reordering allowed. Retry is not considered an error condition, so there is no error logging or reporting done on a retry.

- Split Response

  The 41210 can receive split response for memory reads, and I/O and configuration read and write transactions.

- Target Terminations Initiated by the 41210

  The 41210 responds with a retry to PCI-X when one of the following conditions is met:

  — A memory read transaction occurs and the 41210 delayed transaction queue is full.

  — A LOCK transaction is established from PCI Express*-to-PCI.

  — A memory write transaction occurs and the 41210 has no free buffer space to accept the write.

  — A memory write is from a master other than the master that was previously retried (starvation prevention mechanism).

Note: The 41210 never retries a completion since it always has enough buffer space for all split requests it sends out. No transaction information is retained on any writes.

- Disconnect

  The 41210 disconnects a transaction on PCI-X when one of the following conditions is met:

  — The 41210 cannot accept any more write data and an ADB is reached.

  — A split completion packet being sent, ADB is reached, and the 41210 read buffers are running dry.

  — Inverse decode window ends, and an upstream write is in progress, irrespective of the availability of the write-buffer.

- Target Abort

  The 41210 returns a target abort to PCI-X when one of the following conditions is met:

  — The initial request received a target-abort on the peer PCI bus or a Completion with Completer Abort status on the PCI Express* interface.

  — All requests with address parity error and the parity error response bit is set.

- Master Abort

  — The 41210 master aborts all memory transactions on PCI-X when the bus master enable (BME) bit is cleared.

  — The 41210 returns a master abort response in a split completion when the split cycle master aborted on PCI Express*/peer PCI.

- Split Response

  All memory read cycles that cross the 41210 receive this termination, if they are not retried.

3.3 PCI-X Protocol Specifics

3.3.1 Attributes

Table 16 describes how the 41210 fills in attribute fields where the *PCI-to-PCI Bridge Specification*, Revision 1.1 allows some implementation flexibility.

Table 16. Intel® 41210 Serial to Parallel PCI Bridge Implementation of Requester Attribute Fields

Attribute	Function
No Snoop (NS)	The Intel® 41210 Serial to Parallel PCI Bridge only forwards this attribute in both directions and does nothing with it internally.
Relaxed Ordering (RO)	This bit allows relaxed ordering of transactions, which the 41210 does not permit. This bit is only forwarded in the 41210, and is never generated on PCI-X from an PCI Express* packet or vice-versa.
Tag	Since the 41210 can have two outstanding requests on PCI-X at a time, this field can be either 0 or 1.
Byte counts	From PCI Express*, this attribute is based on the length field from PCI Express*, which is DWord-based.

3.3.2 4 GB and 4 K Page Crossover

The *PCI-X Addendum to the PCI Local Bus Specification*, Revision 1.0b, allows burst transactions to cross page boundaries (in the case of the 41210, this is 4 K) and 4 GB address boundaries. As a PCI-X master, the 41210 always ends the transaction at a 4 K boundary. As a PCI-X target, the 41210 allows a burst beyond a 4 K page boundary. Note that on PCI Express*, neither read nor write requests ever cross a 4 K boundary.

3.3.3 Wait States

The 41210 does not generate wait states as a target.

3.3.4 Split Transactions

- Completer attributes are given in Table 17.

Table 17. Intel® 41210 Serial to Parallel PCI Bridge Implementation of Completer Attribute Fields

Attribute	Function
Byte Count Modified (BCM)	The 41210 does not set this bit.
Split Completion Error (SCE)	The 41210 sets this bit only in the following circumstances: • when a memory read command from PCI-X master is target aborted on PCI Express* • when the 41210 does a queue discard operation of upstream queues
Split Completion Message (SCM)	This bit shadows the SCE bit.

- Unexpected Split Completions

  The 41210 asserts **DEVSEL#** and discards the data when the Requester ID matches the bridge but the tag does not match that of any outstanding requests (0 or 1) from this device.

- Split Completion Messages

  The 41210 can generate error messages only for cycles that cross the bridge that master- or target-abort. At this point, Dword cycles cross the bridge that requires completion (in other words, I/O cycles). Therefore, the 41210 can generate only a “PCI-X Bridge Error” completion message for the memory read commands, as indicated in Table 18.

Table 18. Split Completion Abort Registers

Index	Message
00h	**Master-Abort:** The 41210 encountered a Master-Abort on the destination bus.
01h	**Target-Abort:** The 41210 encountered a Master-Abort on the destination bus

3.4 Arbitration

The 41210 supports a high-performance internal PCI arbiter that supports up to five external masters on each PCI segment. The request inputs into the internal arbiter include five external request inputs and one internal request input. All request inputs to the internal arbiter are split into two groups: a high priority group and a low priority group. Any master, including the internal master, can be programmed to be in either of the two groups. The request inputs into the arbiter can be in one single group. Within a group, priority is round-robin. The entire low-priority group represents one slot in the high-priority group. The 41210 provides a 16-bit arbiter control register to control two aspects of the internal arbiter behavior:

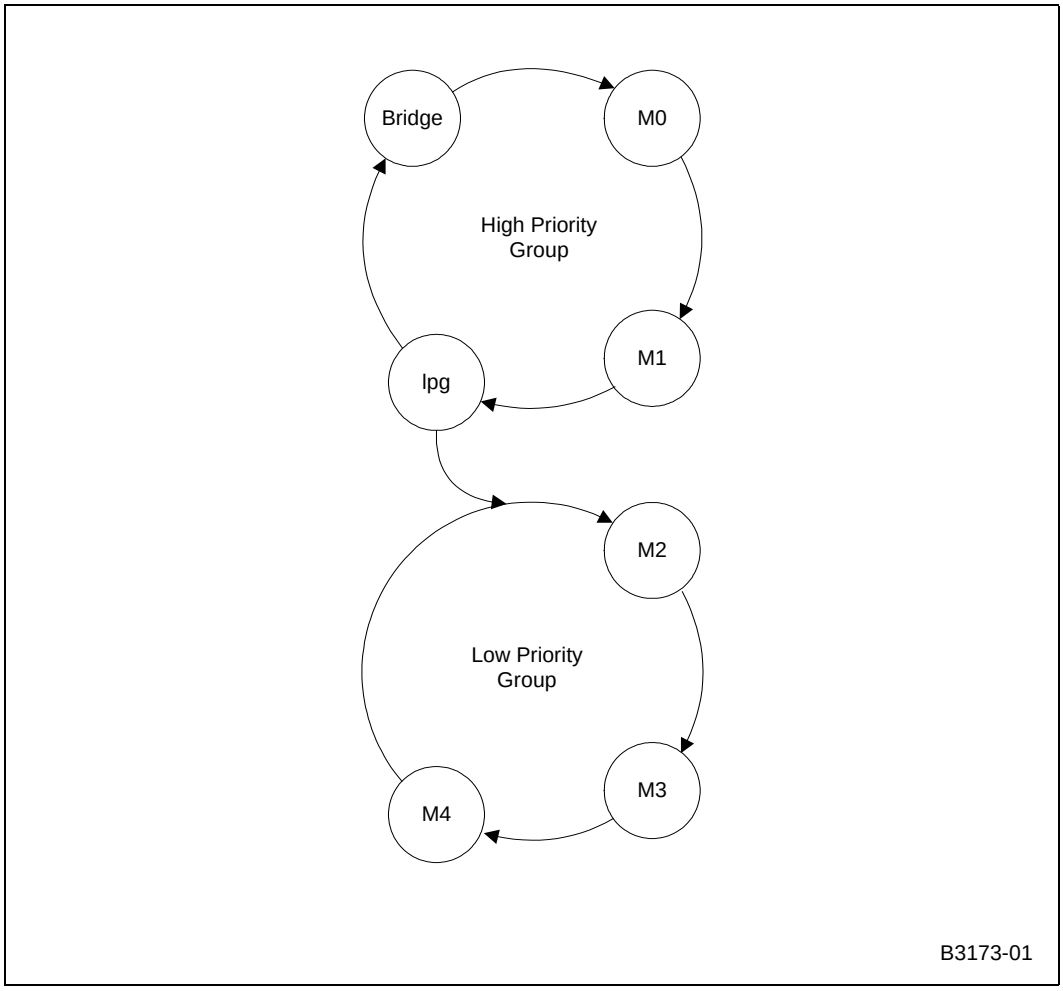
- Priority group for a master (in other words, whether a master is in the low-priority group or the high-priority group)
- Bus parking on last PCI agent or the bridge

For controlling the priority level, there is one bit for each of the PCI REQ# inputs and one bit for the internal request input. Bit[7] in the control register is for the bridge, bit[5] is for REQ[5]#, bit[4] is for REQ[4]#, and so on. A value of 1 in a bit position puts the corresponding master in the high-priority group.

Figure 1 represents the arbiter scheme with bits[7:0] in the arbiter control register set to “110 0011”. In Figure 1, M0 represents master 0 (REQ[0]#), M1 represents master 1 (REQ[1]#) and so on. Bit[8] in the arbiter control register controls the bus parking behavior of the internal arbiter. A value of 0 instructs the internal arbiter to always park the bus on the bridge. A value of 1 instructs the internal arbiter to park the bus on the last PCI master.

The 41210 also supports an 8-bit MTT (Multi-Transaction Timer) register that influences the behavior of the internal arbiter. This register controls the amount of time that the arbiter allows a PCI initiator to perform multiple back-to-back transactions on the PCI bus. The number of clocks programmed in the MTT represents the minimum length of time on the PCI bus that the master is granted the bus as long as the REQ of that master is asserted, before the GNT is given to the next master.

Figure 1. Internal Arbitration Scheme



§ §

Power Management 4

4.1 Hardware-Controlled Active State Power Management

PCI Express* defines a hardware-initiated power management of the PCI Express* Link called active state power management. Under hardware control, the link can be put into a low-power L0s link state or an even lower-power L1 link state. The Intel® 41210 Serial to Parallel PCI Bridge (also called the 41210 Bridge or the 41210) supports only the PCI Express* active state power management link state L0s. The 41210 does ***not*** support optional active state power management link state L1. Active state power management is entirely dependent on traffic and is not initiated by software. The software, however, can enable and disable the active state management by means of the capability structure.

Note: Due to the unreliable behavior described in the *Intel 41210 Serial to Parallel PCI Bridge Specification Update*, Errata #19, L0s active state power management is not supported.

Refer to the *PCI Express* Specification*, Revision 1.0a for more details on the active state power management.

Note: Link state L1 is not supported for hardware-driven active state power management. However, link state L1 is supported for software-driven power management.

4.2 Software-Driven PCI-PM 1.1–Compatible Power Management

The 41210 supports PCI Express* link states L0, L1, and L3, as required to implement PCI-PM 1.1–compatible device states (D0, D3hot, D3cold). When both bridge segments in the 41210 are programmed to the D3hot state, the PCI Express* link enters the link state L1. When the PCI Express* link is in L1 because of software-driven power management, the only message that can cause the link to come out of L1 is a PME message.

Refer to the *PCI Express* Specification*, Revision 1.0a for details of the protocol involved in transitioning the link to the L1 state.

The 41210 also supports the PM_TURN_OFF/PM_TO_ACK protocol to support D3cold/L3 device/link states.

4.3 PCI Bus Power Management

The 41210 supports bus states B0 and B3 corresponding to the bridge device states D0 and D3cold. The 41210 does ***not*** support stopping the PCI bus clocks when in D3hot state and hence does ***not*** support the bus state B2.

4.4 Intel® 41210 Serial to Parallel PCI Bridge Device Power Management

Each bridge segment supports PCI-PM 1.1 device power management states D0, D3hot, and D3cold. Each function, when programmed to the D3hot state, behaves as follows:

- The function responds to configuration cycles from PCI Express*.
- The function initiates and accepts PCI Express* completion transactions.
- The function does ***not*** respond to memory cycles on PCI Express*.
- The function does ***not*** respond to I/O cycles on PCI Express*.
- The function does ***not*** initiate PCI Express* request transactions.
- The function does ***not*** reset its registers, when programmed to D0 from D3hot.
- The 41210 does ***not*** assert PCIRST# when in the D3hot state.

4.5 Power-Management Event Signaling

The 41210 supports conveying PCI power-management events (PME#) over PCI Express* by means of an in-band mechanism. Power-management events are generated on behalf of PCI devices that require a change in their power state. The 41210 does not support any method to "wake" the PCI Express* hierarchy before it signals a PME message (in other words, the 41210 supports neither the WAKE# side band signal nor the in-band tone-generation mechanism). Waking is needed when the upstream component is in a non-communicative state with clock and/or power removed. The expectation for this component is that both ends of the link are fully powered and clocked (in other words, the link is fully communicative) when signaling the PCI power-management events.

The 41210 supports a PME# event pin for conveying power-management events that occur on the secondary PCI bus segments of the 41210. The PME output from all the PCI devices on the segment are wire-ORed to obtain a composite PME signal which is routed to the 41210. The 41210 converts the level-sensitive PME# signal into a PCI Express* message. This message carries the bus number of the PCI bus that caused the PME# assertion. The power-manager software needs the bus-number information when invoked.

Note: Since the bus number of the PCI bus must be passed in the PME_MSG, this scheme functions correctly only for waking from the PCI buses directly below the 41210.

The exact mechanism for generating the PME_MSG packet in the 41210 involves sending a message over PCI Express* whenever the PME input pin is asserted. Note that this packet must carry the bus number of the PCI bus generating the PME#. This means that the 41210 must construct the requestor ID of the PME_MSG packet with the secondary bus-number register in the corresponding PCI-to-PCI Bridge header space. There is a chance that PME messages could be lost. Thus the 41210 implements a counter to periodically sample the PME# pin and generate a message. Refer to the PCI Express* specifications for more details. This polling mechanism creates spurious interrupts to the power-manager software, and the power manager must be able to handle this.

To support the PCI Express* power-management event-signaling protocol, the 41210 supports the following messages:

- PME_Turn_Off
- PME_TO_ACK

PME_Turn_Off is used to turn off PME generation from all PCI Express* devices before the system power-manager disconnects the power from the PCI Express* link hierarchy. The 41210 acknowledges the reception of a PME_Turn_Off message with a PME_TO_ACK message to the north device. Refer to the *PCI Express* Specification*, Revision 1.0a for more details on the PME handshake mechanism.

The 41210 does not use Vaux power. As a result, the 41210 does not support PME# assertion from the D3cold state.

§ §

THIS PAGE INTENTIONALLY LEFT BLANK

Addressing 5

5.1 Addressable Spaces within the Intel® 41210 Serial to Parallel PCI Bridge

Before discussing all the addressing/configuration aspects of the Intel® 41210 Serial to Parallel PCI Bridge (also called the 41210 Bridge or the 41210), this section provides a brief summary of the addressable spaces within the 41210 PCI Express*-to-PCI Bridges (functions 0 and 2), the corresponding access mechanism, and a description of when they are applicable. A detailed description of each of these spaces follows in the later chapters.

Table 19. Addressable Spaces within the Intel® 41210 Serial to Parallel PCI Bridge

Addressable Space	Access			
	PCI Express*	PCIA	PCIB	SMBus
PCI-to-PCI Bridge A Configuration Space (Function 0)	Yes (Type 0)	Yes (Type 0)	No	Yes
PCI-to-PCI Bridge B Configuration Space (Function 2)	Yes (Type 0)	No	Yes (Type 0)	Yes

5.2 Secondary PCI Devices

Devices on the secondary PCI bus can be configured as private devices and hidden from BIOS and host software. Devices are hidden by inhibiting the assertion of the IDSEL input of the device during configuration cycles. This feature is configured through the BINIT register. Public and private devices are supported according to Table 20.

Table 20. Secondary PCI Device Addressing

Device Number	Signal Used for IDSEL Input	Public/Private	Notes
0	AD16	Reserved	Dedicated for bridge
1	AD17	Public or Private	Based on the device-hiding enable bit (bit[2] of the BINIT register) Available for secondary PCI devices
2	AD18		
3	AD19		
4	AD20		
5	AD21		
6	AD22		
7	AD23		
8	AD24	Reserved	Used to address extended configuration space of bridge Based on the upstream-configuration enable bit (bit[1] of the BINIT register)
9	AD25		
10 (0xA)	AD26		
11 (0xB)	AD27		
12 (0xC)	AD28	Public	Available for secondary PCI devices
13 (0xD)	AD29		
14 (0xE)	AD30		
15 (0xF)	AD31		

5.3 Configuration-Space Access

The 41210 supports configuration-space accesses from PCI Express* using both the legacy *PCI-to-PCI Bridge Specification*, Revision 2.3 access mechanism and the enhanced PCI Express* configuration-space access mechanism. For local initialization, the 41210 also supports configuration-pace accesses from the SMBus port and secondary PCI bus.

5.3.1 PCI Express* Configuration Access

The *PCI-to-PCI Bridge Specification*, Revision 1.1 defines the configuration-space region of a PCI function to be up to 256 B. PCI Express* extends the PCI configuration space from 256 B to 4 K. The region up to 256 B can be accessed using the mechanism for configuration accesses defined in the *PCI-to-PCI Bridge Specification*, Revision 1.1. The region above 256 B is accessible only by means of the enhanced configuration access mechanism defined in PCI Express*. This mechanism utilizes a flat memory-mapped address region to access the configuration space. The core-logic chipset converts the legacy *PCI-to-PCI Bridge Specification*, Revision 1.1 or the enhanced PCI Express* configuration-space accesses into PCI Express* configuration cycles.

The extended address bits used to access the configuration region above 256 B are all 0s when the access mechanism compatible with the *PCI-to-PCI Bridge Specification*, Revision 1.1 is used, or when accessing devices on PCI. Note that 41210, when it translates Type 1 configuration transactions from PCI Express*-to-PCI and finds the extended address bits to be non-zero, terminates the transaction with an unsupported request response on PCI Express*. All configuration accesses on PCI Express* are aligned DWORD only.

- Type 0 accesses to the 41210:

  The bridge configuration spaces are accessed from PCI Express* by a Type 0 configuration transaction. Type 0 transactions from PCI Express* (not peer PCI or SMBus) to the bridge segments return a configuration retry response on PCI Express* when the "Configuration Cycle Retry" bit is set in the BINIT register ("Offset FCh: BINIT—Bridge Initialization Register" on page 104).

  The 41210 captures the bus and device numbers from Type 0 configuration writes, to its internal functions, from PCI Express*. The 41210 does ***not*** capture the bus/device number from Type 0 configuration writes to its internal functions, from either the PCI segment or SMBus. The captured bus/device number from Type 0 PCI Express* configuration writes are used by the 41210 in forming the Requester ID/Completer ID on PCI Express* and PCI-X requests/completions. Also, the 41210 does not decode the device number field for Type 0 configuration transactions from PCI Express*.

- Type 1 accesses to the 41210:

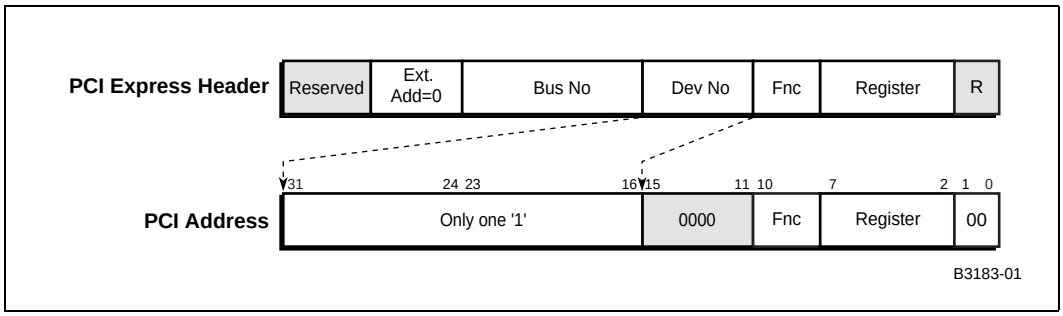
  Type 1 accesses from PCI Express* are intended for the PCI bus only and not for the internal configuration spaces. Type 1 configuration transactions to PCI that do not complete within 40 µs from the time they are received on PCI Express* receive a configuration retry response on PCI Express* when the retry response is enabled by means of the "Bridge Configuration Retry Enable" bit in the PCI Express* Device Control Register ("Offset 4Ch: EXP_DCTL—PCI Express* Device Control Register" on page 93). The 41210 continues to retry the transaction on PCI even when a retry response has been signaled on PCI Express*, and when completed, the transaction is discarded.

- Type 1 to Type 0 translation:

  The 41210 performs a Type 1 to Type 0 translation when the Type 1 transaction is generated on PCI Express* and is intended for a device attached directly to its secondary bus. The bridge must convert the configuration command to a Type 0 format so that the secondary bus device can respond to it. The resulting Type 0 address is driven on the PCI bus, as shown in Figure 2. Device numbers are decoded to assert a single bit (IDSEL) in address bits[31:16]. A device number of 0 converts to PCI AD[16] being a 1; a device number of 2 converts to PCI AD[17] being a 1; and so on. When the device number is greater than 16, all bits (bits[31:16]) are 0. See Table 20, "Secondary PCI Device Addressing" on page 42.

Note: When the device-hiding bit in the BINIT register is set, the 41210 master-aborts all Type 0 transactions to the PCI bus targeting device numbers 0 to 9. Device numbers 10 to 15 are never hidden; in other words, they are never master-aborted.

Figure 2. Type 1 to Type 0 Translation (PCI and PCI-X)



- Type 1-to-Type 1 Forwarding:

  The 41210 passes a Type 1 PCI Express* configuration cycle as a Type 1 configuration cycle on PCI when it is intended for a device attached to a bus below the bridge and beyond the bus directly attached to the secondary side of the bridge.

  The 41210 forwards a Type 1 configuration cycle unchanged to the PCI bus when the Type 1 configuration cycle on PCI Express* has a bus number that falls in the range defined by the lower limit (exclusive) in the secondary bus number register and the upper limit (inclusive) in the subordinate bus number register.

  As an error response, the 41210 returns an “unsupported request” completion when the extended address bits are non-zero.

Note: The device-hiding bit in the BINIT register has no effect when forwarding a Type 1 transaction to PCI as a Type 1 transaction.

- Type 1 to Special Cycle Forwarding

  The 41210 translates a Type 1 configuration write transaction on PCI Express* into a special cycle on PCI, but does not translate a Type 1 configuration access on PCI to a special cycle on PCI Express*. A PCI Express* Type 1 configuration cycle is be converted to a special cycle on the PCI interface when **all** of the following conditions are true:

  — The device number field is equal to 1 1111b.

  — The function number field is equal to 111b.

  — The register number field is equal to 00 0000b.

  — The bus number is equal to the value in the secondary bus number register in configuration space.

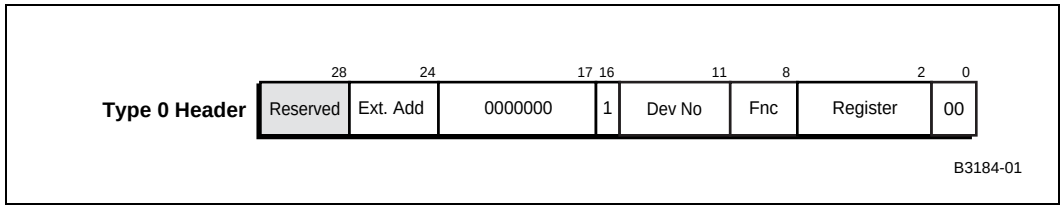
  The address and data are forwarded unchanged. Devices ignore the address and decode only the bus command. The data phase contains the special cycle message. The transaction master-aborts on PCI, but results in a normal completion on the opposite bus (normal completion status on PCI Express*; no DEVSEL# on PCI).

5.3.2 Type 0 Configuration Access from PCI-X Interface

The 41210 supports inbound Type 0 configuration accesses from PCI-X to access registers on the corresponding source bridge segment. Type 0 accesses from PCI-X cannot be used to access registers in any other functions within the 41210 other than the source bridge segment; nor can it be used to access devices upstream of the 41210.

Instead of having a secondary IDSEL# pin, the 41210 reserves a device number of 0 for itself. The 41210 claims a Type 0 configuration transaction from PCI-X when the Upstream Configuration Enable bit is set in the Bridge Initialization Register ("Offset FCh: BINIT—Bridge Initialization Register" on page 104) and the AD[16] signal of the transaction is asserted (HIGH) during the address phase of the configuration transaction. The format for the Type 0 configuration cycle on PCI-X is shown in Figure 3.

Figure 3. Upstream Type 0 PCI-X Configuration Cycle Address Format



The 41210 ignores the device number and function number fields during Type 0 decode. The 41210 does not use the device number in the PCI-X Type 0 configuration transaction to program its internal (primary) device number field in the PCI-X Bridge Status register.

Note: Platforms supporting inbound Type 0 configuration cycles to the 41210 must not use address bit[16] or bits[27:24] for IDSEL on the motherboard.

5.3.3 SMBus Configuration Access

The 41210 provides a mechanism for SMBus access to its internal configuration registers. This mechanism provides a means for server management controllers to access the registers for debugging, and also provides a means for custom configuration of the bridge based on usage models. See Section 8, "System Management Bus Interface" on page 55 for more details.

5.4 I/O Space Access Mechanism

One I/O window can be set up in the PCI-to-PCI Bridge space for forwarding I/O transactions from PCI Express*-to-PCI. Refer to Section 5.6, "VGA Addressing" on page 49 to see how I/O cycles in the VGA range are handled. The registers and register bits listed below define the setup and control of this I/O window:

- I/O Base and Limit (IOBL) registers in the PCI-to-PCI Bridge configuration space
- I/O enable bit (IOSE) in the command register in the PCI-to-PCI Bridge configuration space

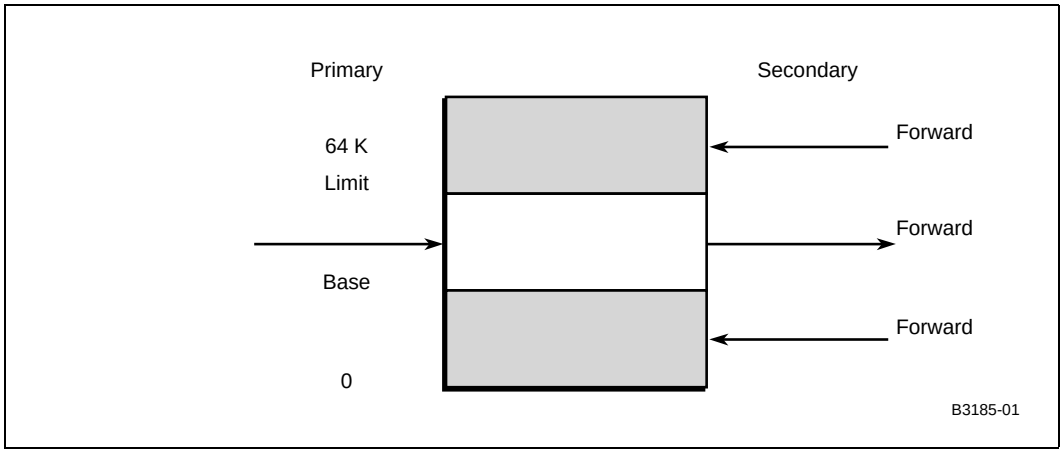
To enable downstream I/O transactions, the I/O enable bit must be set in the command register in the 41210 configuration space (bit 0 at offset 04h–05h). When the I/O enable bit is not set, all I/O transactions initiated on PCI Express* receive a master-abort completion. The 41210 implements one set of I/O base and limit address registers in configuration space that define an I/O address range for the bridge. PCI Express* I/O transactions with addresses that fall inside the range defined by the I/O base and limit registers are forwarded to PCI, and PCI I/O transactions with addresses that fall outside this range are master-aborted.

Setting the base address to a value greater than that of the limit address turns off the I/O range. When the I/O range is turned off, no I/O transactions are forwarded to PCI even when the I/O enable bit is set.

The base register consists of an 8-bit field at configuration address 1Ch, and a 16-bit field at address 30h. The top four bits (bits[7:4] of address 1Ch) of the 8-bit field define bits[15:12] of the I/O base address. The bottom four bits (bits[3:0]) read only as 0h to indicate that the 41210 supports 16-bit I/O addressing only. Bits[11:0] of the base address are assumed to be 0, which naturally aligns the base address to a 4 KB boundary. The I/O base upper 16-bit register at offset 30H is reserved. After chip reset, the value of the I/O base address is initialized to 0000H.

The I/O limit register consists of an 8-bit field at offset 1Dh and a 16-bit field at offset 32h. The top four bits (bits[7:4] of address 1Dh) of the 8-bit field define bits[15:12] of the I/O limit address. The bottom four bits (bits[3:0]) read only as 0h to indicate that 16-bit I/O addressing is supported. Bits[11:0] of the limit address are assumed to be FFFh, which naturally aligns the limit address to the top of a 4 KB I/O address block. The 16 bits I/O base and limit registers at offsets 30h and 32h are not implemented, since the 41210 supports only 16-bit I/O addressing. After chip reset, the value of the I/O limit address is reset to 0FFFh (in other words, the lower 4 K in the 64 K space).

Figure 4. I/O Forwarding



Error Response: I/O transactions from PCI Express* that do not match the I/O address forwarding window of either PCI-to-PCI Bridges results in a UR response. Note that software is responsible for making sure that the I/O window programmed into the registers of the two PCI-to-PCI Bridges do not overlap.

5.5 Memory Space Access Mechanism

The 41210 supports 64 bits of memory addressing on both interfaces.

Two memory windows can be setup for forwarding memory transactions from PCI Express*-to-PCI. These windows are defined as part of the standard PCI-to-PCI Bridge configuration space. Inverse decoding is used for forwarding transactions from PCI-to-PCI Express*. Refer to Section 5.6, “VGA Addressing” on page 49 to see how memory cycles in the VGA range are handled. The registers and register bits listed below control the setup and operation of these memory windows:

- Memory-mapped I/O base and limit (MBL) registers
- Prefetchable memory base and limit (PMBL) registers
- Prefetchable memory base and limit upper 32 bits (PMBLU32) register
- Memory enable (MSE) bit in the command register
- Master enable bit (BME) in the command register

To enable downstream memory transactions, the memory space enable bit in the command register must be set (bit[1] of offset 04h–05h). To enable upstream memory transactions, the master enable bit in the command register must be set (bit[2] of offset 04h–05h). The 41210 does not prefetch data from downstream PCI devices. Upstream prefetching is controllable by settings in the “Offset 178h: PREFCTRL—Prefetch Control Register” on page 119.

5.5.1 Memory-Mapped I/O Window

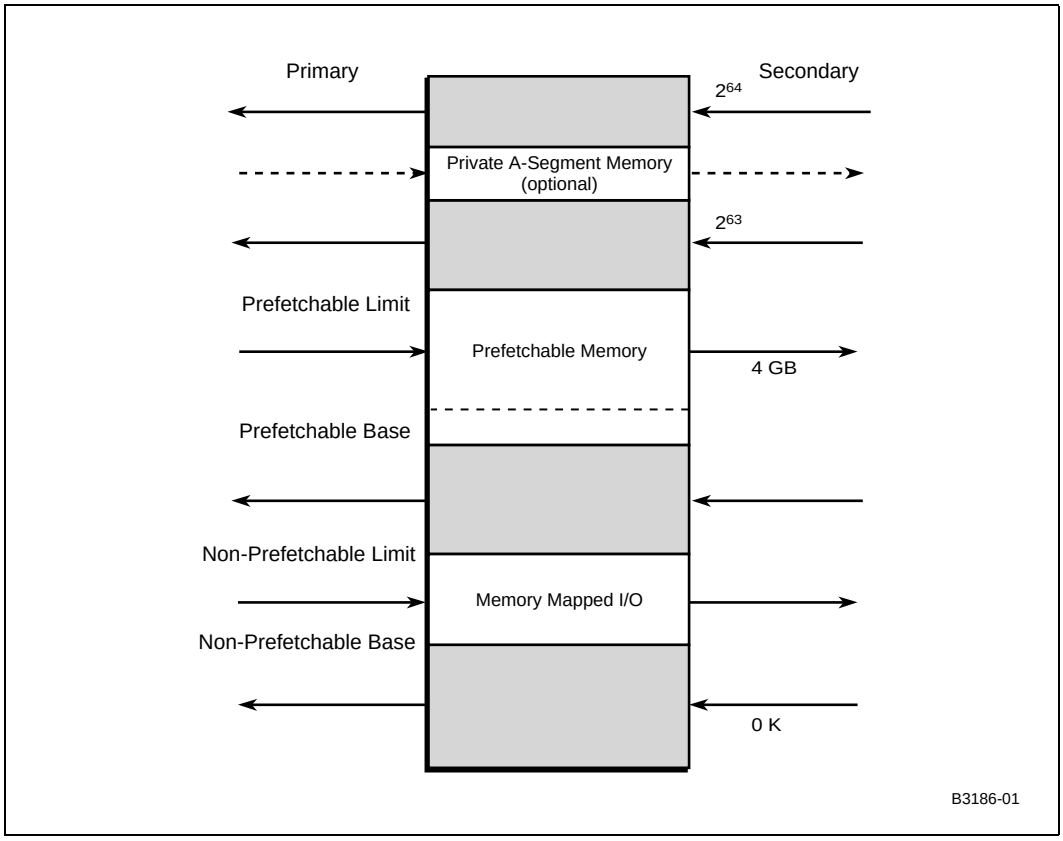
Software uses the memory-mapped I/O window to map all non-prefetchable (in other words, reads that have side effects, such as reads to FIFOs, or “read-to-clear” status registers) memory space into PCI memory space.

The memory-mapped I/O base address register and memory-mapped I/O limit address register define an address range that the bridge uses to determine when to forward memory commands. The 41210 forwards a memory transaction from PCI Express* to PCI when the address falls within the range. The 41210 forwards it from PCI to PCI Express* (or the peer PCI segment) when the address is outside the range and does not fall into the prefetchable memory range. This memory range supports 32-bit addressing only (addresses 4 GB). It has a granularity and alignment of 1 MB.

This range is defined by a 16-bit base address register at offset 20h in the configuration space and a 16-bit limit address register at offset 22h. The most significant 12 bits of each of these registers correspond to bits[31:20] of the memory address. The least significant four bits are hard-wired to 0. The least significant 20 bits of the base address are assumed to be all 0s, which results in a natural alignment to a 1 MB boundary. The least significant 20 bits of the limit address are assumed to be all 1s, which results in an alignment to the top of a 1 MB block.

Note: Setting the base to a value greater than that of the limit turns off the memory range.

Figure 5. Memory Forwarding



5.5.2 Prefetchable Memory Window

The prefetchable memory base and address registers, along with their upper 32-bit counterparts, define an additional address range that the 41210 uses to forward accesses. Software maps the prefetchable PCI memory spaces to this window. The 41210 still treats the memory reads in this region as non-prefetchable. The 41210 forwards a memory transaction from PCI Express* to PCI when the address falls within the range. The 41210 forwards transactions from PCI to PCI Express* (or the peer PCI segment) when the address is outside the range and does not fall into the regular memory range. This memory range supports 64-bit addressing, and has a granularity and alignment of 1 MB.

The least-significant 32 bits of the range are defined by a 16-bit base register at offset 24h in the configuration space and a 16-bit limit register at offset 28h. The most-significant 12 bits of each of these registers correspond to bits[31:20] of the memory address. The least-significant 4 bits are hard-wired to 1h, indicating 64-bit address support. The least-significant 20 bits of the base address are assumed to be all 0s, which results in a natural alignment to a 1 MB boundary. The least-significant 20 bits of the limit address are assumed to be all 1s, which results in an alignment to the top of a 1 MB block.

The most-significant 32-bits of the range are defined by a 32-bit base register at offset 28h in the configuration space, and a 32-bit limit register at offset 2Ch.

Note: Setting the entire base (with the most-significant 32-bits) to a value greater than that of the limit turns off the memory range.

5.5.3 Opaque Memory Window

When the opaque memory window is enabled, the 41210 hard codes certain address ranges to the secondary segment of each bridge.The hard-coded address ranges are as follows:

- A[63:62] = 10 Secondary side of A-Segment
- A[63:62] = 11 Secondary side of B-Segment

These address ranges are not forwarded from the PCI Express* interface to the corresponding secondary side and are also never forwarded from the secondary to the PCI Express* interface, regardless of the setting of the prefetchable base and limit registers.

Note: Even when the opaque memory window is enabled, the normal behavior defined for the BME, MSE, and IOSE bits in the PCICMD register is still applicable.

5.6 VGA Addressing

When a VGA-compatible device exists behind a 41210 bridge, the VGA enable bit in the bridge control register is set (offset 3 at 3Eh–3Fh).

When this bit is set, the 41210 forwards all transactions addressing the VGA frame buffer memory and VGA I/O registers from PCI Express* to PCI, regardless of the values of the 41210 base and limit address registers. When set, the 41210 does not forward VGA frame buffer memory accesses to PCI Express* regardless of the values of the memory address ranges. However, the I/O enable and memory enable bit in the command register must still be set.

When this bit is cleared, the 41210 forwards transactions addressing the VGA frame buffer memory and VGA I/O registers from PCI Express* to PCI when the defined memory and I/O address ranges enable forwarding. When cleared, accesses to the VGA frame buffer memory are forwarded from PCI to PCI Express* when the defined memory address ranges enable forwarding. However, the master enable bit must still be set. The VGA I/O addresses are never forwarded to PCI Express* when the upstream I/O enable bit in BINIT register is cleared. When this bit is set and also the VGA enable bit is set, the 41210 does not forward the VGA I/O addresses from PCI to PCI Express*.

The VGA frame buffer consists of the memory address range 000A 0000h–000B FFFFh.

The VGA I/O addresses consist of the I/O addresses 3B0h–3BBh and 3C0h–3DFh. These I/O addresses are aliased every 1 KB throughout the first 64 KB of I/O space, when the VGA 16-bit decode bit in the bridge control register (bit[4]) is cleared. This means that address bits[9:0] (3B0h–3BBh and 3C0h–3DFh) are decoded, bits[15:10] are not decoded and can be any value, and address bits[31:16] must be all 0s. When the VGA 16-bit decode bit is set, the 41210 does the entire 16-bit decode on the VGA I/O addresses. When software sets the VGA enable bit in one bridge, the ISA enable bit must be set in the other bridge.

§ §

Transaction Ordering 6

The Intel® 41210 Serial to Parallel PCI Bridge (called hereafter the 41210 Bridge or 41210) follows the producer-consumer model of a standard PCI Express*-to-PCI bridge. Based on this model, the 41210 implements a set of ordering rules in the upstream and downstream directions. The ordering plane covered by these rules spans the transaction domain covered by PCI Express* and either of the two PCI segments. The 41210 uses a single PCI Express* virtual channel, for both PCI segments.

Accesses to the internal 41210 configuration registers follow no ordering relationship with respect to transactions moving to and from PCI and PCI Express* buses. This means that upstream memory and configuration reads and writes (when enabled via the BINIT register) are completed out of order with the transactions pending in the 41210 upstream queues towards PCI Express*. Downstream memory/configuration transactions to the internal register space can complete out of order with respect to transactions pending in the downstream queues towards PCI. Software must be aware that any semaphore mechanism implemented through the internal 41210 register space requires a dummy read to the PCI or PCI Express* space to push the writes that are pending in the 41210 queues in either direction. The ordering tables in the next two sections do not consider these transactions.

6.1 Upstream Transaction Ordering

Table 21 lists the combined set of ordering rules in the upstream path of the 41210 for PCI transactions.

Table 21. Upstream Transaction Ordering

Row Pass Column	Posted Write	Delayed/Split Read Request	Delayed/Split Read Completion	Delayed/Split Write Completion
Posted write	No	Yes	No	No
Delayed/split read request	No	Yes[1]	Yes	No
Delayed/split write request	No	Yes	Yes	No
Delayed/split read completion	No	Yes	Yes	No
Delayed/split write completion	No	Yes	Yes	No

NOTE:
1. Subsequent requests only (prefetches). All upstream initial requests are in order.

6.2 Downstream Transaction Ordering

Table 22 lists the combined set of ordering rules in the downstream path of the 41210.

Table 22. Downstream Transaction Ordering

Row pass Column	Posted Write	Delayed/Split Read Request	Delayed/Split Write Request	Delayed/Split Read Completion
Posted write	No	Yes	Yes	Yes
Delayed/split read request	No	Yes[1]	Yes	Yes
Delayed/split write request	No	Yes	Yes	Yes
Delayed/split read completion	No	Yes	Yes	Yes
Delayed/split write completion	No	Yes	Yes	Yes

NOTE:
1. The Intel® 41210 Serial to Parallel PCI Bridge supports two downstream completion required requests per PCI segment. Downstream delayed/split read requests can pass each other when issued on the PCI bus.

6.3 Relaxed Ordering/No-Snoop Support

The 41210 forwards the PCI Express*/PCI-X relaxed ordering and no-snoop attributes to PCI-X/PCI Express*. No internal optimization is done with the relaxed ordering attribute.

§ §

Interrupt Support 7

The Intel® 41210 Serial to Parallel PCI Bridge (called hereafter the 41210 Bridge or 41210) can generate an in-band interrupt request on PCI Express* for boot devices and for systems that do not support Message Signaled Interrupts (MSI).

7.1 Legacy Interrupt Sharing

PCI Express* provides interrupt messages that emulate the legacy wired mechanism. This feature allows I/O devices to signal PCI-style interrupts using a pair of ASSERT and DEASSERT messages. This message pairing preserves the level-sensitive semantics of the PCI interrupts on PCI Express*.

The four virtual wire interrupts (INTA–INTD) correspond to the four interrupt wires defined in the *PCI Local Bus Specification*, Revision 2.3. The 41210 routes its PCI interrupt pins and the internal interrupts to the PCI Express* INTx interrupts as shown in Table 23.

Table 23. INTx Routing Table

A_INTx# Interrupt Pins	B_INTx# Interrupt Pins	PCI Express* INTx Message
A_INTA#	B_INTA#	INTA
A_INTB#	B_INTB#	INTB
A_INTC#	B_INTC#	INTC
A_INTD#	B_INTD#	INTD

Each row in Table 23 indicates a logical ORing of the interrupts in that row. The ASSERT/DEASSERT message sent out on PCI Express* captures the asserting/deasserting edge of the signal that represents the logical OR of the interrupts in that row.

The 41210 uses its primary bus number and device number in the Requester ID field for the PCI Express* INTx messages. As stated in the *PCI Express* Specification*, Revision 1.0a, the function number is reserved for interrupt messages and is always 0.

Note: PCI Express* Assert_INTx/Deassert_INTx messages are not inhibited by the Bus Master Enable (BME) bit.

7.2 Interrupt Routing for Devices behind a Bridge

Given the legacy interrupt sharing scheme shown in Table 23, to get the best legacy interrupt performance (by reducing interrupt sharing), adapter boards must select the appropriate INTA#–INTD# input pin to use on each PCI bus segment. The chosen interrupt input also imposes a PCI device number requirement for the interrupt source as shown in Table 24.

Table 24. Interrupt Binding for Devices behind a Bridge

Device Number on Secondary Bus	Interrupt Pin on Device	Interrupt Pin on the Intel® 41210 Serial to Parallel PCI Bridge
0[1], 4, 8[2], 12, 16, 20, 24, 28	INTA#	INTA#
	INTB#	INTB#
	INTC#	INTC#
	INTD#	INTD#
1, 5, 9[2], 13, 17, 21, 25, 29	INTA#	INTB#
	INTB#	INTC#
	INTC#	INTD#
	INTD#	INTA#
2, 6, 10[2], 14, 18, 22, 26, 30	INTA#	INTC#
	INTB#	INTD#
	INTC#	INTA#
	INTD#	INTB#
3, 7, 11[2], 15, 19, 23, 27, 31	INTA#	INTD#
	INTB#	INTA#
	INTC#	INTB#
	INTD#	INTC#

NOTES:
1. Device number 0 is reserved for the bridge and must not be assigned to secondary devices.
2. AD[27:24], which correspond to devices[11:8], must not be used for IDSEL# connections, because these signals are used when accessing the extended configuration space in the bridge from the secondary bus.

§ §

System Management Bus Interface 8

The SMBus interface allows the Intel® 41210 Serial to Parallel PCI Bridge (called hereafter the 41210 Bridge or 41210) to serve as a slave device residing on the SMBus for system management functions and provides for full access to the configuration registers in each function. The SMBus implementation has the following characteristics:

- Is based on the *System Management Bus Specification*, Revision 2.0 (SMBus)
- Allows for slave-mode operation only
- Provides full read/write access to internal configuration and memory spaces

The SMBus address is set up on PERST# by sampling the SMBUS pins. When the pins are sampled, the resulting address is assigned as shown in Table 25:

Table 25. SMBus Address Assignments

Bit	Value
7	1
6	1
5	SMBUS[5]
4	0
3	SMBUS[3]
2	SMBUS[2]
1	SMBUS[1]

8.1 SMBus Commands

The 41210 supports six SMBus commands:

- Block Write
- Block Read
- Word Write
- Word Read
- Byte Write
- Byte Read

Sequencing these commands initiates accesses to the internal configuration and memory registers. For high reliability, the 41210 also supports the optional packet-error-checking feature (CRC-8) and is enabled or disabled with each transaction.

Every configuration and memory read or write consists first of an SMBus write sequence that initializes the bus number, device, function number, register offset, and so on. The term sequence is used because these variables can be initialized by the SMBus master with a single block write or multiple word or byte writes. The last write in the sequence that completes the initialization performs the internal configuration/memory read or write. The SMBus master can then initiate a read sequence, which returns the status of the internal read or write command and also the data in case of a read.

Each SMBus transaction has an 8-bit command driven by the master. The command encodes the information shown in Table 26:

Table 26. SMBus Command Encoding

Bit	Description
7	**Begin:** When set, this bit indicates the first transaction of the read or write sequence.
6	**End:** When set, this bit indicates the last transaction of the read or write sequence.
5	**Reserved:** Must be set to 0.
4	**PEC Enable:** When set, indicates that PEC is enabled. When set, each transaction in the sequence ends with an extra CRC byte. CRC is checked on writes and generated on reads. PEC does not include the command byte itself.
3:2	**Internal Command:** 00 Read Dword 01 Write Byte 10 Write Word 11 Write Dword All accesses are naturally aligned to the access width. This field specifies the command to be issued by the SMBus slave logic to the internal registers.
1:0	SMBus **command:** 00 Byte 01 Word 10 Block 11 Reserved This field specifies the SMBus command to be issued on the SMBus. This field is used as an indication of the length of transfer so that the slave knows when to expect the PEC packet (when enabled).

8.2 Initialization Sequence

All configuration read and writes are accomplished through SMBus write(s) followed by an SMBus read (for a read command). For configuration access, the SMBus write sequence is used to initialize the following parameters:

- Bus number
- Device/function number
- 12-bit register number (in two separate bytes on SMBus)

Each of the parameters above is sent on the SMBus in separate bytes. The register number parameter is initialized with two bytes, and the 41210 ignores the most significant four bits of the second byte that initializes the register number.

The initialization of the information can be accomplished through any combination of the supported SMBus write commands (Block, Word or Byte). The internal command field for each write must specify the same internal command every time (read or write). After all the information is set up, the last write (end bit is set) initiates an internal read or write command. On an internal read, when the data is not available before the slave interface acknowledges this last write command (ACK), the slave “clock stretches” until the data returns to the SMBus interface unit. On an internal write, when the write is not complete before the slave interface acknowledges this last write command (ACK), the slave “clock stretches” until the write completes internally. When an error occurs (internal time-out or internal abort) during the internal access, the last write command receives a NACK.

8.2.1 Configuration

The 41210 supports only read Dword to internal register space. All configuration reads are accomplished through an SMBus write (or writes) and are followed later by an SMBus read to read the status and the read data. For SMBus read transactions, the last byte of data (or the PEC byte when enabled), is NACKed by the master to indicate the end of the transaction. The SMBus read command returns the status of the previous internal command and the data associated previous internal read command. The status field encoding is shown in Table 27:

Table 27. SMBus Status Byte Encoding

Bit	Description
7	Internal time-out. This bit is set when an SMBus request is not completed in 2 ms internally.
6	Reserved
5	Internal master abort
4	Internal target abort
3:1	Reserved
0	Successful

Examples of configuration reads are shown in Figure 6 through Figure 9. For the definition of the diagram conventions below, refer to the *System Management Bus Specification*, Revision 2.0.

Figure 6. DWord Configuration Read Protocol (SMBus Block Write/Block Read, PEC Enabled)

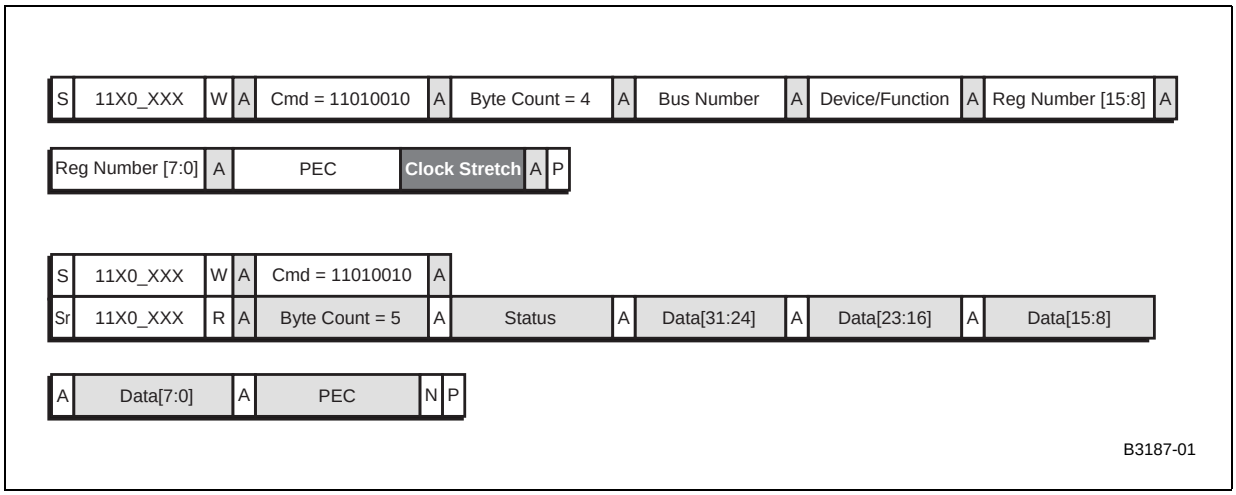


Figure 7. DWord Configuration Read Protocol (SMBus Word Write/Word Read, PEC Enabled)

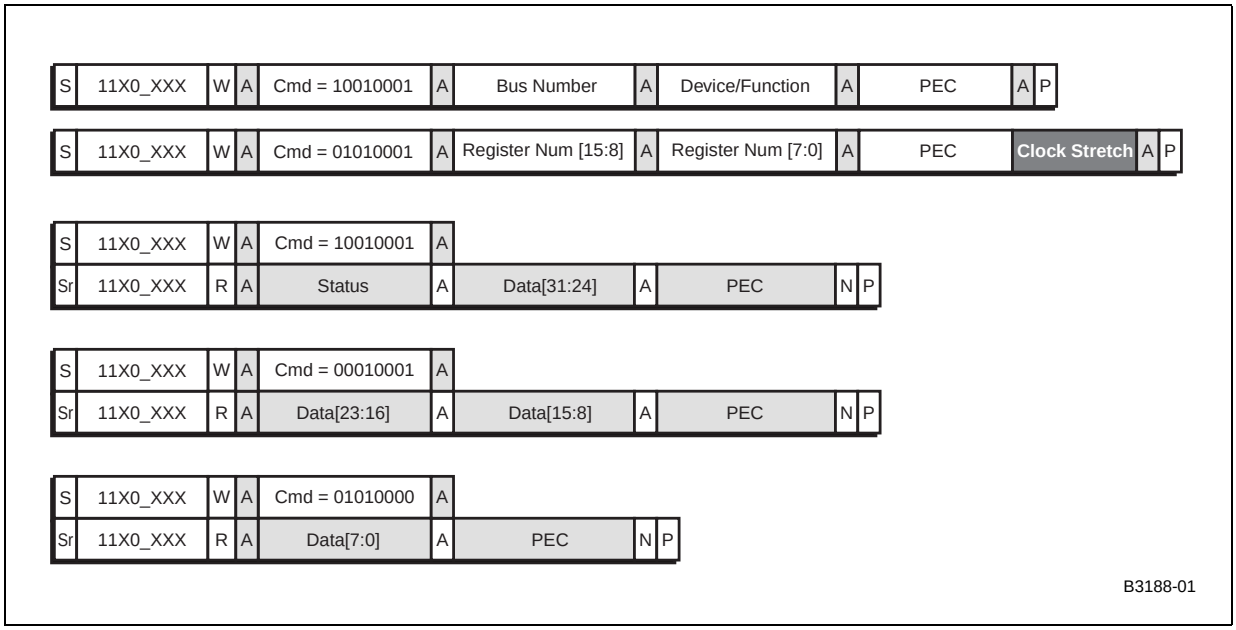


Figure 8. DWord Configuration Read Protocol (SMBus Block Write/Block Read, PEC Disabled)

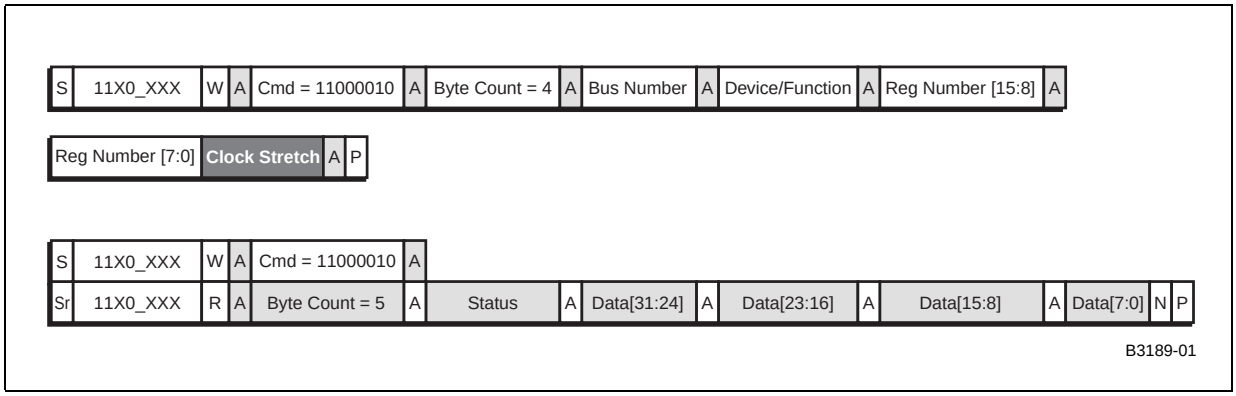
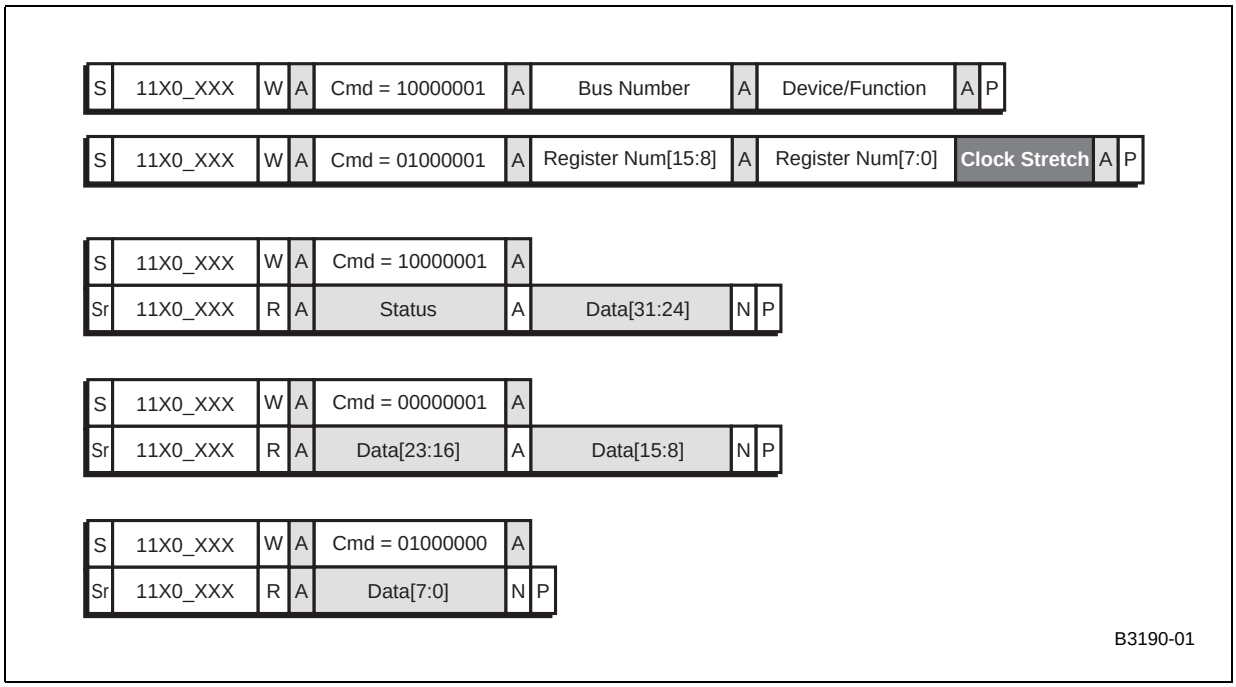


Figure 9. DWord Configuration Read Protocol (SMBus Word Write/Word Read, PEC Disabled)



8.2.2 Configuration Writes

Configuration writes are accomplished through a series of SMBus writes. As with reads, a write sequence is used first to initialize the bus number, device, function, and register number for the configuration access. The writing of this information can be accomplished through any combination of the supported SMBus write commands (Block, Word or Byte).

Note: On SMBus, there is no concept of byte enables. Therefore, the register number written to the slave is assumed to be aligned to the length of the internal command. In other words, for a *Write Byte* internal command, the register number specifies the byte address. For a *Write DWord* internal command, the two least-significant bits of the register number are ignored. This is different than PCI; with PCI, the byte enables are used to indicate the byte of interest.

After all the information is set up, the SMBus master initiates one or more writes which set up the data to be written. The final write (end bit is set) initiates an internal configuration. The slave interface can potentially "clock-stretch" the last data write until the write completes without error. When an error occurs, the SMBus interface NACKs the last write operation just before the stop bit. Examples of configuration writes are illustrated in Figure 10 and Figure 11. All the figures are shown with PEC enabled. When PEC is disabled, there is no PEC byte in any of the sequences, and the PEC enable bit in the command field is 0.

For the definition of the diagram conventions used in Figure 10 and Figure 11, refer to the *System Management Bus Specification*, Revision 2.0.

Figure 10. DWord Configuration Write Protocol (SMBus Block Write, PEC Enabled)

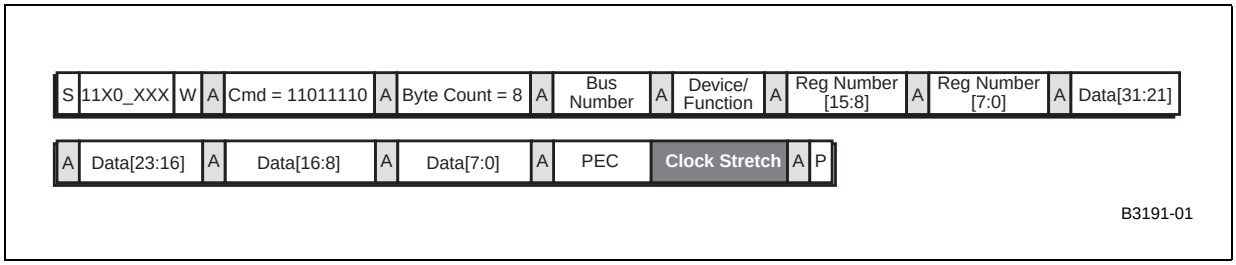
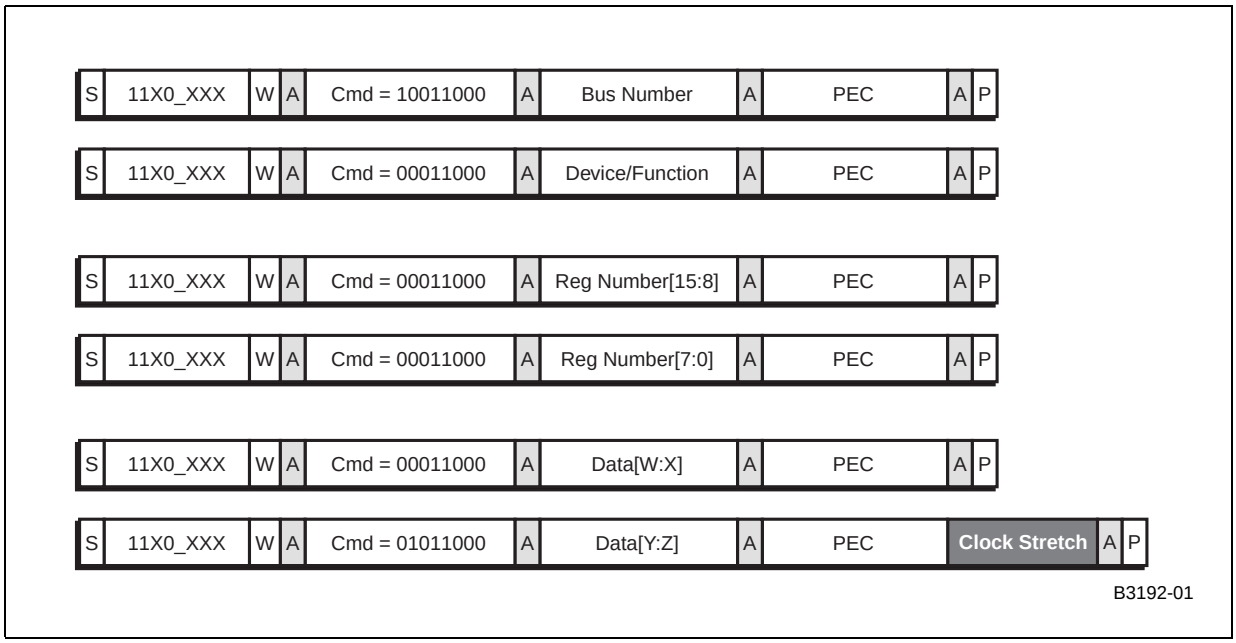


Figure 11. DWord Configuration Write Protocol (SMBus Byte Write, PEC Enabled)



8.3 Error Handling

The SMBus slave interface handles two types of errors: internal and PEC.

Internal errors can occur when the target function is busy servicing a request from the PCI Express* interface. The SMBus unit may time-out these transactions and return a NACK for the read or write command. Additionally, an internal error can occur when the read or write command receives a master or target abort on the internal interface. When the master receives a NACK, the entire transaction must be reattempted.

When the master supports packet error checking (PEC), and the PEC enable bit in the command is set, the PEC byte is checked in the slave interface. When the check indicates a failure, the slave NACKs the PEC packet and does not issue the command on the internal interface.

Note: An SMBus master must either do PEC on all transactions in a sequence or not do it at all. PEC cannot be disabled in the middle of a sequence. A PEC error in the middle of a sequence must be re-started from the beginning of the sequence that set the begin bit.

8.4 SMBus Interface Reset

The master has two ways to reset the slave interface state machine in the 41210:

- The master holds **SCLK** low for 25 ms cumulative. “Cumulative” in this case means that all the “low time” for **SCLK** is counted between the start and stop bit. When this count totals 25 ms before reaching the stop bit, the interface is reset.
- The master holds **SCLK** continuously high for 50 ms.

Besides these methods, the SMBus interface in the 41210 is also reset on a **PERST#**, **RSTIN#**, or an in-band warm reset from PCI Express*.

§ §

Local Initialization 9

The Intel® 41210 Serial to Parallel PCI Bridge (called hereafter the 41210 Bridge or 41210) includes device-specific registers that allow for control of the bridges's behavior, both internally and externally. Examples of these device-specific registers are the arbiter control register, the prefetch control register, and so on. Depending on the usage model, these registers might need to be programmed to a different value than the reset-default, every time 41210 goes through a component reset.

When application-specific initialization of the 41210 is required, the CFGRETRY strap must be asserted at the rising edge of PERST#. This places the 41210 Bridge in a local initialization mode, and all configuration accesses from PCI Express* is retried by returning a completion with the configuration request retry status. As soon as the local configuration is completed, the "Local Initialization In Progress bit" in the "Offset FCh: BINIT—Bridge Initialization Register" on page 104 must be cleared to enable host access to the bridge configuration registers.

Local initialization can be accomplished via SMBus access or Type 0 configuration cycles from the secondary bus. The CFGRST# output is asserted whenever the configuration space is reset and can be used as a control signal to re-initialize the application-specific parameters. The X_RST# signal must not be used since the configuration space is not cleared due to a software-initiated secondary bus reset.

Device-specific registers are listed below:

- "Offset 40h: BCNF—Bridge Configuration Register" on page 90
- "Offset 42h: MTT—Multi-Transaction Timer" on page 91
- "Offset 43h: PCLKC—PCI Clock Control" on page 91
- "Offset FCh: BINIT—Bridge Initialization Register" on page 104
- "Offset 16Ah: ARB_CNTRL—Internal Arbiter Control Register" on page 117
- "Offset 170h: SSR—Strap Status Register" on page 118
- "Offset 178h: PREFCTRL—Prefetch Control Register" on page 119

§ §

THIS PAGE INTENTIONALLY LEFT BLANK

Clock and Reset 10

10.1 Clocking

The Intel® 41210 Serial to Parallel PCI Bridge (called hereafter the 41210 Bridge or the 41210) always uses the PCI Express* REFCLK as its primary clock input and drives the PCI clock outputs. The clock domains are shown in Table 28:

Table 28. Clock Domains

Clock Domain	Frequency	Source	Usage
PCI Express*	100 MHz Differential	External	PCI Express* differential clocks
PCI	133/100/66/33 MHz	Internal	PCI bus clock outputs
SMBus	10–100 KHz	Source Synchronous	This pin is controlled by the driver of the SMBus interface, and typically runs between 10 and 100 KHz
TCK	0–16 MHz	External	JTAG clock

10.2 Device Reset

Four types of resets can be performed on the 41210. These are listed from the highest-level reset to the lowest-level reset:

- **PERST#:** This signal indicates stable power when high and causes an asynchronous reset of the entire chip when low. This signal must be connected to the PERST# pin on the PCI Express* connector.
- **RSTIN#:** When asserted, this signal causes an asynchronous reset of the 41210. This reset is used for debugging purposes only and must be pulled high for normal operation.
- PCI Express* **Reset:** This reset is a message coming on the PCI Express* interface and resets all configuration registers with the exception of sticky bits.
- **Software PCI Reset:** This reset is initiated by writing to the bridge control register of the PCI configuration space. This reset affects only the associated bridge segment and is commonly referred to as the Secondary Bus Reset (SBR).

These resets are described in more detail in the following sections.

10.2.1 PERST# Reset Mechanism

All the voltage sources in the system are tracked by a system component that asserts the PERST# signal only after all the voltages have been stable for some predetermined time. The 41210 receives the PERST# signal as an asynchronous input, meaning that there is no assumed relationship between the assertion or the de-assertion of PERST# and the reference clock. While the PERST# is de-asserted, the 41210 holds all logic in reset.

The PERST# reset clears all internal state machines and logic, and initializes all registers to their default states, including "sticky" error bits that are persistent through all other reset classes. To eliminate potential system-reliability problems, all devices are also required to either tristate their outputs or to drive them to safe levels during such a power-on reset.

The 41210 keeps PCIRST# asserted for a minimum of 320 ms after the deassertion of PERST#.

Refer to the *PCI Express* Specification*, Revision 1.0a for details of the relationship between PERST# assertion and the stability of the clocks and power at the inputs of the 41210.

10.2.2 RSTIN# Reset Mechanism

As soon as the system is up and running, a full system reset may be required to recover from system-error conditions related to various device or subsystem failures. The RSTIN# reset mechanism is a hot-reset mechanism that accomplishes this recovery without clearing the "sticky" error-status bits which track the cause of the error conditions of the device or subsystem.

A hot reset can be initiated by asserting the RSTIN# signal. This signal is treated as an asynchronous input to the 41210, meaning that there is no assumed relationship between the host reference clock and the assertion or the de-assertion of RSTIN#.

When the 41210 goes through a reset due to RSTIN# assertion, the link goes down, which is interpreted by the upstream component as a surprise extraction which may cause system instability.

The 41210 keeps PCIRST# asserted for a minimum of 320 ms after the deassertion of RSTIN#.

10.2.3 PCI Express* Reset Mechanism

There is no reset signal on the PCI Express*, and all reset communication is in-band. The upstream PCI Express* device communicates the fact that it is entering and coming out of a reset using messages. The 41210 responds by also going through a reset. In accordance with the PCI Express* protocol, this incoming message is asynchronous to the reference clock.

When the upstream device puts the 41210 Bridge in reset through the in-band reset mechanism, the 41210 resets its core and PCI interfaces. Sticky bits are reserved

The 41210 keeps PCIRST# asserted for a minimum of 320 ms after the deassertion of the PCI Express* in-band reset message.

10.2.4 Software PCI Reset (SBR—Secondary Bus Reset)

Commonly referred to as the Secondary Bus Reset (SBR), the software PCI reset is initiated by a write to the bridge control register and resets only the particular PCI segment. This reset can be used for various reasons, including but not limited to the following:

- Recovering from error conditions on the secondary bus
- Redoing enumeration
- Changing the operating frequency of the bus (33/66/100/133 MHz)
- Changing the operating mode of the bus (PCI or PCI-X)

This reset is synchronous to the PCI clock domain in which it is used. SBR is strictly restricted to the particular PCI segment and affects neither the other PCI segment nor the rest of the 41210 Bridge logic. Writes to the bridge configuration register with a new frequency or bus mode have no effect until the SBR is completed (see “Offset 40h: BCNF—Bridge Configuration Register” on page 90).

§ §

THIS PAGE INTENTIONALLY LEFT BLANK

Error Handling 11

For each interface, the Intel® 41210 Serial to Parallel PCI Bridge (called hereafter the 41210 Bridge or 41210) implements the specified error-logging and escalation actions as per the interface rules. For example, errors encountered on the PCI interface follow the logging and escalation rules of PCI. The error escalation mechanisms implemented by the 41210 can be fully masked. This feature provides the platform software with the ability to pick and choose what it wants to do on any of the error conditions. All logging registers specific to the 41210 are “sticky” (these registers retain their values) through any chip reset except a PERST# reset cycle.

11.1 PCI Express* Errors

The 41210 supports the PCI Express* advanced error-reporting capability, which allows for system-level error recovery and debugging. The capability includes both the base error-reporting features and the bridge-specific extensions for reporting PCI and PCI-X errors.

PCI Express* errors are classified as either correctable errors or uncorrectable errors:

- Correctable errors are those for which hardware exists to correct the errors.
- Uncorrectable errors are errors for which hardware does not exist to correct the errors.

Uncorrectable errors are further classified into fatal and non-fatal errors, with non-fatal errors indicating an unreliable link.

PCI Express* supports three different error messages to support these error classes: ERR_COR, ERR_UNC and ERR_FATAL. Refer to the *PCI Express* Specification*, Revision 1.0a for details of the various PCI Express* errors and how they are signaled and escalated.

PCI Express* error logging specifies a set of advanced transaction-logging registers as an added capability.

11.2 PCI Errors

PCI and PCI-X errors include several sources of error, such as the following:

- Address errors
- Data-parity errors
- Split-completion errors
- Master aborts
- Target aborts

Some of these errors are fatal and some are non-fatal. The PCI-X specifications specify a set of rules for the behavior of a bridge under a variety of error conditions that could happen on the bus. To aid the system software/driver in error recovery and debugging, the 41210 implements those rules on PCI along with the error-logging and routing control specific to the 41210.

11.2.1 Error Types

PCI errors are classified into two categories: fatal and non-fatal:

- Fatal errors are those that have the potential to cause data corruption. Software must be careful to contain and escalate these errors (when needed).
- Non-fatal errors are those that do not cause any data corruption. These errors include driver errors such as master-abort on PCI and target errors such as target-abort.

All errors on PCI are uncorrectable and are forwarded to PCI Express* as such.

The fatal class of errors includes:

- Data parity errors on PCI
- Address and attribute parity errors on PCI

The non-fatal class of errors includes:

- Target Aborts on PCI
- Master Aborts on PCI

11.2.2 Termination of Completion Required Transactions

11.2.2.1 Immediate Termination on the PCI-X Interface

An immediate termination occurs when the 41210 Bridge masters a transaction on PCI or PCI-X and receives an “immediate termination” response for that transaction. Table 29 describes the completion-status translation for immediate terminations. The behavior described for completion-required cycles is independent of the setting of the Master Abort Mode bit, and is also independent of whether the cycle is exclusive (locked) or not.

Table 29. Completion-Status Translation for Immediate Terminations

PCI-X Termination		PCI Express* Completion
Normal completion		Successful
Normal completion with data parity error	Memory, I/O, configuration reads	Successful with poisoned TLP
	Configuration, I/O writes[1]	UR
	Configuration write to special cycle conversion[2]	Successful
Master abort		UR
Target abort		CA

NOTES:
1. In PCI mode, the Intel® 41210 Serial to Parallel PCI Bridge samples PERR# asserted and generates the UR completion.
2. PERR# is not signaled for a special cycle data parity error. SERR# is asserted instead.

11.2.2.2 Split Termination on PCI-X Interface

A split-termination error translation occurs when a completion-required transaction receives a "split termination" response when originally mastered on the PCI-X bus, and a "split completion" error message is later received for the original request. Table 30 describes the completion-status translation for PCI-X split-completion terminations. The behavior described in Table 30 is independent of the Master Abort Mode bit and whether or not the cycle is exclusive (locked).

Note: When a target or master abort is returned on PCI Express* for the first read of an exclusive access, the secondary PCI-X bus is not locked. This is especially important in regard to the completion messages "byte count out of range, "device specific", and "reserved/invalid codes". The 41210 Bridge does not lock its bus on these errors, even though they are not explicitly master- or target-aborts on the PCI-X interface.

Table 30. Completion-Status Translation for PCI-X Split-Completion Terminations

PCI-X Split Termination	Message		PCI Express* Completion Status
	Class	Index	
Successful	0	00h	Successful
Master abort	1	00h	UR
Target abort	1	01h	CA
Write data parity error	1	02h	UR
Byte count out of range	2	00h	UR
Write data parity error	2	01h	UR
Device-specific	2	8Xh	CA
Reserved/invalid	Others		CA

11.2.2.3 Split Termination on PCI Express* Interface

Table 31 shows the split-completion errors received on the PCI Express* interface and how they translate to PCI-X.

Table 31. Completion-Status Translation for PCI Express* Split-Completion Terminations

PCI Express* Completion Status	PCI Completion	
Successful (SC)	Successful	
Unsupported Request (UR)	MAM = 1	Memory reads: PCI target abort[1]
		I/O reads: PCI target abort
		I/O writes: PCI target abort
	MAM = 0	Memory reads: PCI return all Fs
		I/O reads: PCI return all Fs
		I/O writes: Normal completion
	PCI-X split master abort[2]	
Completer Abort (CA)	PCI target abort[1] PCI-X split target abort[1]	

NOTES:
1. Data is returned to the point of error, and then a target abort is signaled.
2. The 41210 Bridge issues a split-completion error message and either master aborts or target aborts the remaining completion sequence when an abort is detected on the PCI Express*/peer interface. If several bytes of data return successfully for the request, the data is returned to the point of error (rounded to the nearest ADB), and then the split-completion error message is generated.

§ §

Register Description 12

This chapter describes the registers of the Intel® 41210 Serial to Parallel PCI Bridge.

12.1 Register Nomenclature and Access Attributes

Table 32 describes the nomenclature used for describing bit attributes throughout this chapter.

Table 32. Bit Attribute Definitions

Mnemonic	Attribute
RO	**Read-Only:** This bit cannot be altered by software. This bit can be hard-wired to return a fixed value at all times, or it can be set by hardware on an event.
RsvdP	**Reserved and Preserved:** These bits are reserved for future RW implementations; software must preserve the value read for writes to bits. The Intel® 41210 Serial to Parallel PCI Bridge hardware implements these bits as read-only 0s.
RsvdZ	**Reserved and Zero:** These bits are reserved for future RWC implementations; software must use 0 for writes to bits. The 41210 hardware implements these bits as read-only 0s.
ROS	**Read-Only Sticky:** These bits are read-only and cannot be altered by software. The bits are not cleared by reset and can be reset only with the **PERST#** reset condition.
RZSet	**Read Zero to Set:** Reading this bit when the current value of the bit is 0 causes the bit to flip to a 1. Software must write a 1 to clear this bit. Writing a 0 has no effect.
RW	**Read-Write:** Software can do a full read and write of this bit.
RW1Set	**Read and Write One to Set:** Software must write a 1 to set this bit. Writing a 0 has no effect on this bit. Software can clear this bit through a separate RWC bit, or it can be reset by hardware.
RWS	**Read-Write and Sticky:** Software can read and write this bit. The bit can be reset only by a **PERST#** reset.
RWC	**Read and Write One to Clear:** When this bit is set, software must write a 1 to this bit to clear it. Writing a 0 has no effect.
RWCS	**Read and Write One to Clear and Sticky through reset:** When this bit is set, software must write a 1 to this bit to clear it. Writing a 0 has no effect. The bit can be reset only by a **PERST#** reset.
Strap	**Strap:** This is a read-only register. The power-on default is based on sampling a strap pin at the rising edge of **PERST#**.
WT	**Write Transient:** This bit is always read as a 0. Writing a 1 to this bit causes other side-effects that are specific to every WT bit.

Note: Software must not attempt to write to the registers that are marked “reserved”. Writing to these registers yields undetermined results. Reads of these registers can yield either value. Note that the behavior of individual register bits that are marked “reserved” is in accordance with the attribute definition for that bit.

12.2 Configuration Registers

The bridge configuration space follows the standard PCI Express*-to-PCI Bridge configuration space format. Refer to the *PCI Express*-to-PCI Bridge Specification*, Revision 1.0a for details on the format. Each 41210 Bridge contains an identical set of registers as described in this section for its respective PCI segment.

Table 33 and Table 34 show the configuration registers of the 41210 and their address byte offset values. Figure 12 presents the capabilities supported by the 41210.

Figure 12. Intel® 41210 Serial to Parallel PCI Bridge Capabilities

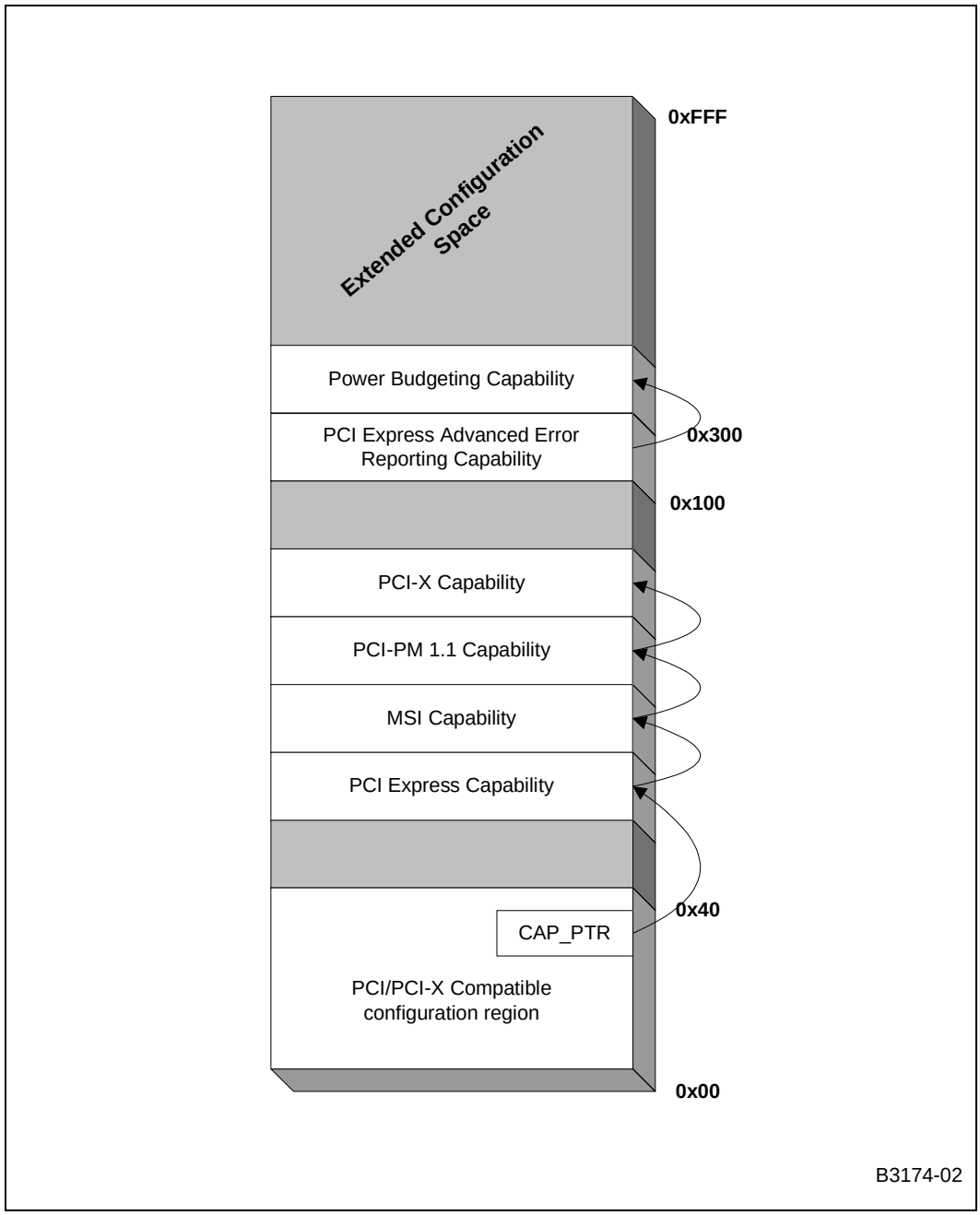


Table 33. Legacy Configuration Space

	Byte Offset
DID / VID	00h
PSTS / PCICMD	04h
Class Code (CC) / REVID	08h
Reserved / HEADTYP / PMLT / CLS	0Ch
Reserved	10h
Reserved	14h
SMLT / BNUM	18h
SSTS / IOBL	1Ch
MBL	20h
PMBL	24h
PMBU32	28h
PMLU32	2Ch
IOBLU16	30h
Reserved / CAPP	34h
Reserved	38h
BCTRL / INTR	3Ch
PCLKC / MTT / BCNF	40h
EXP_CAP / EXP_NXTP / EXP_CAPID	44h
EXP_DCAP	48h
EXP_DSTS / EXP_DCTL	4Ch
EXP_LCAP	50h
EXP_LSTS / EXP_LCTL	54h
Reserved	58h
MSI_MC / MSI_NXTP / MSI_CAPID	5Ch
MSI_MA	60h
	64h
Reserved / MSI_MD	68h
PM_PMC / PM_NXTP / PM_CAPID	6Ch
PM_DATA / PM_BSE / PM_PMCSR	70h
Reserved	74h
	78h
	7Ch

	Byte Offset
Reserved	80h
	84h
	88h
	8Ch
	90h
	94h
	98h
	9Ch
	A0h
	A4h
	A8h
	ACh
	B0h
	B4h
	B8h
	BCh
	C0h
	C4h
	C8h
	CCh
	D0h
	D4h
PX_SSTS / PX_NXTP / PX_CAPID	D8h
PX_BSTS	DCh
PX_USTC	E0h
PX_DSTC	E4h
Reserved	E8h
	ECh
	F0h
	F4h
	F8h
BINIT	FCh

Table 34. PCI Express* Extended Configuration Space

Register	Byte Offset
EXPAERR_CAPID	100
ERRUNC_STS	104
ERRUNC_MSK	108
ERRUNC_SEV	10C
ERRCOR_STS	110
ERRCOR_MSK	114
ADVERR_CTL	118
HDR_LOG	11C 120 124 128
Reserved / PCIXERRUNC_STS	12C
Reserved / PCIXERRUNC_MSK	130
Reserved / PCIXERRUNC_SEV	134
Reserved / PCIXERRUNC_PTR	138
PCIXHDR_LOG	13C 140 144 148
Reserved	14C–167
ARB_CNTRL / Reserved	168
Reserved / Reserved	16C
Reserved / SSR	170
Reserved / Reserved	174
PREFCTRL	178 17C
Reserved	180–2FF
PWRBGT_CAPID	300
Reserved / PWRBGT_DSEL	304
PWRBGT_DATA	308
Reserved	30C–3FF

12.2.1 Offset 00h: ID—Identifiers

Contains the vendor and device identifiers for software.

Table 35. Offset 00h: ID—Identifiers

Bits	Type	Reset		Description
31:16	RO	A	B	**Device ID (DID):** These bits indicate the device number assigned by Intel to the Intel® 41210 Serial to Parallel PCI Bridge.
		0340h	0341h	
15:0	RO	8086h		**Vendor ID (VID):** This 16-bit field indicates that Intel is the vendor.

12.2.2 Offset 04h: PCICMD—Command Register

This register controls how the device behaves on the primary interface (PCI Express*). As this component is a bridge, additional command information is located in a separate Bridge Control register ("Offset 3Eh: BCTRL—Bridge Control" on page 88).

Table 36. Offset 04h: PCICMD—Command Register (Sheet 1 of 2)

Bits	Type	Reset	Description
15:11	RO	00h	Reserved
10	RW	0b	**INTx Mask:** The bridge does not generate internal interrupts. The value of this bit has no meaning.
9	RO	0b	**Fast Back-to-back enable (FBE):** This bit has no meaning on PCI Express*. This bit must be hard-wired to 0.
8	RW	0b	**SERR# Enable (SEE):** This bit enables reporting of non-fatal and fatal errors to the root complex. 0 = Disable reporting errors 1 = Enable reporting of non-fatal and fatal errors to the root complex **NOTE:** Errors are reported when enabled either through this bit or through the PCI Express*-specific bits in the Device Control Register ("Offset 4Ch: EXP_DCTL—PCI Express* Device Control Register" on page 93).
7	RO	0b	**Wait Cycle Control (WCC):** Reserved
6	RW	0b	**Parity Error Response Enable (PERE):** This bit controls the setting of the master data parity error bit in the Status Register ("Offset 06h: PSTS—Primary Device Status" on page 79) in response to a parity error received on the PCI Express* interface (poisoned TLP) or peer PCI interface. 0 = The 41210 ignores these errors on the PCI Express*/peer-PCI interface. 1 = The 41210 reports read completion data parity errors on PCI Express* and sets the MDPD bit in the status register.
5	RO	0b	**VGA Palette Snoop Enable (VGA_PSE):** Reserved
4	RO	0b	**Memory Write and Invalidate Enable (MWIE):** Memory write and invalidate transactions are not generated, since PCI Express* does not have a corresponding transfer type.
3	RO	0b	**Special Cycle Enable (SCE):** Reserved

Table 36. Offset 04h: PCICMD—Command Register (Sheet 2 of 2)

Bits	Type	Reset	Description
2	RW	0b	**Bus Master Enable (BME):** This bit controls the ability of the 41210 to issue memory and I/O read/write requests on the PCI Express* interface. 0 = The 41210 does not respond to any memory or I/O transactions on the PCI interface and stops issuing new requests on PCI Express*. 1 = The 41210 processes transactions normally. **NOTE:** This bit does not stop completions on PCI Express* from being issued. Software must ensure that all upstream posted transactions are flushed in the bridge segment when this bit is set. Otherwise, delayed completions (such as configuration read completions) can be stuck behind a posted write and cannot proceed from PCI to PCI Express*.
1	RW	0b	**Memory Space Enable (MSE):** This bit controls the response of the 41210 when the 41210 is the target of a memory transaction from a primary or secondary interface. 0 = Every memory transaction targeting a secondary interface is master-aborted, and every memory transaction from secondary to primary is claimed. 1 = Primary-to-secondary and secondary-to-primary forwarding follows the normal rules for memory forwarding.
0	RW	0b	**I/O Space Enable (IOSE):** This bit controls the response of the 41210 when the 41210 is the target of I/O transactions from primary or secondary interfaces. 0 = Every I/O transaction targeting secondary is master-aborted, and every memory transaction from secondary to primary is claimed, provided that the upstream I/O enable bit in the BINIT register is also set. 1 = Primary-to-secondary and secondary-to-primary forwarding follows the normal rules for memory forwarding.

12.2.3 Offset 06h: PSTS—Primary Device Status

For the writable bits in this register, writing a 1 clears the bit. Writing a 0 to the bit has no effect.

Table 37. Offset 06h: PSTS—Primary Device Status (Sheet 1 of 2)

Bits	Type	Reset	Description
15	RWC	0b	**Detected Parity Error (DPE):** This bit is set when a poisoned TLP is received from PCI Express* or a data parity error is detected from the peer PCI segment (writes or read completions). This bit is set even when the parity error response enable bit (bit[6] of the PCICMD Register—"Offset 04h: PCICMD—Command Register" on page 78) is not set. 0 = No error 1 = Poisoned TLP received or Data Parity Error detected
14	RWC	0b	**Signaled System Error (SSE):** This bit is set when ERR_FATAL or ERR_NONFATAL messages are sent to the root complex and the SERR enable bit in the PCICMD Register ("Offset 04h: PCICMD—Command Register" on page 78) is set. 0 = No error 1 = ERR_FATAL or ERR_NONFATAL message sent
13	RWC	0b	**Received Master Abort (RMA):** This bit is set when the 41210 receives a completion with "Unsupported Request Completion" status on the PCI Express* interface. 0 = No error 1 = "Unsupported Request Completion" status received on PCI Express* interface
12	RWC	0b	**Received Target Abort (RTA):** This bit is set when the 41210 receives a completion with "Completer Abort" (CA) status on the PCI Express* interface. 0 = No error 1 = Completer Abort (CA) status received on PCI Express* interface

Table 37. Offset 06h: PSTS—Primary Device Status (Sheet 2 of 2)

Bits	Type	Reset	Description
11	RWC	0b	**Signaled Target Abort (STA):** This bit is set when a completion packet with Completer Abort (CA) status is generated on PCI Express*. 0 = No error. 1 = Completer Abort (CA) status transmitted on PCI Express* interface
10:9	RO	00b	**DEVSEL# Timing (DVT):** These bits have no meaning on PCI Express*. Fast decode timing is reported.
8	RWC	0b	**Master Data Parity Error Detected (MDPD):** This bit is set when the 41210 detects an uncorrectable data error. This bit is set when the parity error response enable bit (PERE) in the Command Register ("Offset 04h: PCICMD—Command Register" on page 78) is set and one of the following conditions occurs: • The bridge receives a completion with a poisoned TLP on the PCI Express* interface. • The bridge receives a completion with poisoned data on the peer PCI segment. • The bridge poisons a write request on the PCI Express* interface. 0 = No error 1 = Data Parity Error in completion packet received or a write request transmitted
7	RO	0b	**Fast Back-to-Back Capable (FBC):** This bit has no meaning on PCI Express*.
6	RO	0b	Reserved
5	RO	0b	**66 MHz Capable (C66):** This bit has no meaning on PCI Express*.
4	RO	1b	**Capabilities List Enable (CAPE):** This bit indicates the capabilities pointer in the 41210. Offset 34H indicates the offset for the first entry in the linked list of capabilities. 0 = No capabilities enabled 1 = Capabilities enabled and accessible from 34H
3	RO	0b	**Interrupt Status:** The 41210 does not generate internal interrupts. This bit is hard-wired to 0.
2:0	RO	0h	Reserved

12.2.4 Offset 08h: REVID—Revision ID

This register is the Revision ID Register.

Table 38. Offset 08h: REVID—Revision ID

Bits	Type	Reset	Description
7:0	RO	00h	**Revision ID (RID):** These bits indicate the stepping (die version) of the Intel® 41210 Serial to Parallel PCI Bridge. 0000 0000 A-0 stepping

12.2.5 Offset 09h: CC—Class Code

This register contains the class code, sub-class code, and programming interface for the device.

Table 39. Offset 09h: CC—Class Code

Bits	Type	Reset	Description
23:16	RO	06h	**Base Class Code (BCC):** The value of 06h indicates that this is a bridge device.
15:8	RO	04h	**Sub Class Code (SCC):** This 8-bit value indicates that this device is a PCI-to-PCI Bridge.
7:0	RO	00h	**Programming Interface (PIF):** This bit indicates that this device is standard (non-subtractive) PCI-to-PCI Bridge.

12.2.6 Offset 0Ch: CLS—Cache-Line Size

This register indicates the cache-line size of the system.

Table 40. Offset 0Ch: CLS—Cache Line Size

Bits	Type	Reset	Description
7:0	RW	00h	**Cache Line Size (CLS):** These bits specify the system cache-line size in units of Dwords: • 08h: 32-byte line (8 DWords) • 10h 64-byte line • 20h 128-byte line Any value outside this range defaults to a 64-byte line. When creating read requests to PCI Express*, this value is used to partition speculative PCI read requests on cache-line–aligned boundaries. This register has no other effect on the 41210.

12.2.7 Offset 0Dh: PMLT—Primary Master Latency Timer

This register does not apply to PCI Express*, and is maintained as R/W for software compatibility.

Table 41. Offset 0Dh: PMLT—Primary Master Latency Timer

Bits	Type	Reset	Description
7:3	RO	00h	**Time Value (TV):** Not applicable for PCI Express*
2:0	RO	000b	Reserved

12.2.8 Offset 0Eh: HEADTYP—Header Type

This register determines how the rest of the configuration space is laid out.

Table 42. Offset 0Eh: HEADTYP—Header Type

Bits	Type	Reset	Description
7	RO	1b	**Multi-function device (MFD):** Reserved as 1 to indicate that the 41210 is a multi-function device.
6:0	RO	01h	**Header Type (HTYPE):** These bits define the layout of addresses 10h through 3Fh in the configuration space. These bits read as 01h to indicate that the register layout conforms to the standard PCI-to-PCI Bridge layout.

12.2.9 Offset 18h: BNUM—Bus Numbers

This register contains the primary, secondary, and maximum subordinate bus number registers.

Table 43. Offset 18h: BNUM—Bus Numbers

Bits	Type	Reset	Description
23:16	RW	00h	**Subordinate Bus Number (SBBN):** These bits indicate the highest PCI bus number downstream of this bridge. Every Type 1 configuration cycle on PCI Express* with a bus number greater than the secondary bus number and less than or equal to the subordinate bus number is forwarded as a Type 1 configuration cycle to the secondary PCI bus.
15:8	RW	00h	**Secondary Bus Number (SCBN):** These bits indicate the bus number of the PCI device to which the secondary interface is connected. Any Type 1 configuration cycle matching this bus number is translated to a Type 0 configuration cycle and run on the PCI bus.
7:0	RW	00h	**Primary Bus Number (PBN):** These bits indicate the PCI Express* bus number. Any Type 1 configuration cycle with a bus number less than this number is not accepted by this bridge (in other words, it may still match the other bridge).

12.2.10 Offset 1Bh: SMLT—Secondary Master Latency Timer

This timer controls the amount of time that the 41210 continues to burst data on its secondary interface. The counter starts counting down from the assertion of FRAME#. When the grant is removed, then the expiration of this counter results in the de-assertion of FRAME#. When the grant is not removed, then the 41210 may continue ownership of the bus. The secondary master latency timer default value is 64 in PCI-X mode (see Section 8.6.1 of the *PCI-X Addendum to the PCI Local Bus Specification*, Revision 1.0b).

Table 44. Offset 1Bh: SMLT—Secondary Master Latency Timer

Bits	Type	Reset	Description
7:3	RW	00h: PCI 40h: PCI-X	**Secondary Latency Timer (TV):** This 5-bit value indicates the number of PCI clocks, in 8-clock increments, during which the bridge remains as a master of the PCI bus when another master is requesting use of the PCI bus.
2:0	RO	000b	Reserved

12.2.11 Offset 1Ch: IOBL—I/O Base and Limit

This register defines the base and limit, aligned to a 4 KB boundary, of the I/O area of the bridge. Accesses from PCI Express* that are within the ranges specified in this register are sent to PCI when the I/O space enable bit is set. Accesses from PCI Express* that are outside the ranges specified result in an Unsupported Request response.

Table 45. Offset 1Ch: IOBL—I/O Base and Limit

Bits	Type	Reset	Description
15:12	RW	0h	**I/O Limit Address Bits [15:12] (IOLA):** These bits define the top address of an address range to determine when to forward I/O transactions from PCI Express* to PCI. These bits correspond to address lines[15:12] for 4 KB aligned window. Bits[11:0] are assumed to be FFFh.
11:8	RO	0h	**I/O Limit Addressing Capability (IOLC):** Each of these bits is hard-wired to 0, indicating support for 16-bit I/O addressing only.
7:4	RW	0h	**I/O Base Address Bits [15:12] (IOBA):** These bits define the bottom address of an address range to determine when to forward I/O transactions from one interface to another. These bits correspond to address lines[15:12] for 4 KB alignment. Bits[11:0] are assumed to be 000h.
3:0	RO	0h	**I/O Base Addressing Capability (IOBC):** Each of these bits is hard-wired to 0, indicating support for 16-bit I/O addressing only.

12.2.12 Offset 1Eh: SSTS—Secondary Status

For the writable bits in this register, writing 1 to the bit clears the bit. Writing 0 to the bit has no effect.

Table 46. Offset 1Eh: SSTS—Secondary Status

Bits	Type	Reset	Description
15	RWC	0b	**Detected Parity Error (DPE):** This bit is set to 1 whenever the bridge detects an address or data parity error on the PCI bus. This bit is set even when the Parity Error Response Enable bit of the Bridge Control Register (bit[0], "Offset 3Eh: BCTRL—Bridge Control" on page 88) is not set.
14	RWC	0b	**Received System Error (RSE):** This bit is set to 1 when a SERR# assertion is received on PCI.
13	RWC	0b	**Received Master Abort (RMA):** This bit is set to 1 whenever the bridge, as an initiator on the PCI bus, receives a master-abort, or when the bridge receives a PCI-X split completion packet with a master-abort.
12	RWC	0b	**Received Target Abort (RTA):** This bit is set to 1 whenever the bridge, as an initiator on PCI, receives a target-abort on PCI. For "completion required" PCI Express* packets, this event forces a completion status of "target abort" on PCI Express*, and sets the Signaled Target Abort in the Primary Status Register ("Offset 06h: PSTS—Primary Device Status" on page 79).
11	RWC	0b	**Signaled Target Abort (STA):** This bit is set to 1 when the bridge, as a target on the PCI bus, signals a target abort.
10:9	RO	01b	**DEVSEL# Timing (DVT):** These bits indicate that the 41210 responds in medium decode time to transactions on the PCI interface (secondary bus).
8	RWC	0b	**Master Data Parity Error Detected (MDPD):** This bit is set to 1 when all of the following are true: • The bridge is the initiator on PCI. • PERR# is detected to be asserted. • The Parity Error Response Enable bit in the Bridge Control Register (bit[0], "Offset 3Eh: BCTRL—Bridge Control" on page 88) is set. This bit is also set when the 41210 receives a split-completion message from PCI-X, which indicates a write data parity error (regardless of the setting of the Parity Error Response Enable bit). Refer to *PCI-X Addendum to the PCI Local Bus Specification*, Revision 1.0b for details.
7	RO	1b	**Fast Back-to-Back Capable (FBC):** This bit indicates that the secondary interface can receive fast back-to-back cycles.
6	RO	0b	Reserved
5	RO	1b	**66 MHz Capable (C66):** This bit indicates that the secondary interface of the bridge is 66 MHz-capable.
4:0	RO	00h	Reserved

12.2.13 Offset 20h: MBL—Memory Base and Limit

Defines the base and limit, aligned to a 1 MB boundary, of the non-prefetchable memory area of the bridge. Accesses from PCI Express* that are within the ranges specified in this register are sent to PCI when the Memory Space Enable bit is set. Accesses from PCI that are outside the ranges specified are forwarded to PCI Express* when the Bus Master Enable bit is set.

Note: Even though this region is non-prefetchable, peer reads from PCI can potentially prefetch through this window. This prefetching can be turned off with the Prefetch Policy bits (PP bits[42:41], "Offset 178h: PREFCTRL—Prefetch Control Register" on page 119).

These registers are cleared to all 0s on reset.

Note: This register must be programmed appropriately to enable or disable the space.

Table 47. Offset 20h: MBL—Memory Base and Limit

Bits	Type	Reset	Description
31:20	RW	000h	**Memory Limit (ML):** These bits are compared with bits[31:20] of the incoming address to determine the upper 1 MB-aligned value (exclusive) of the range. The incoming address must be less than this value.
19:16	RO	0h	Reserved
15:4	RW	000h	**Memory Base (MB):** These bits are compared with bits[31:20] of the incoming address to determine the lower 1 MB-aligned value (inclusive) of the range. The incoming address must be greater than or equal to this value.
3:0	RO	0h	Reserved

12.2.14 Offset 24h: PMBL—Prefetchable Memory Base and Limit

This register defines the base and limit, aligned to a 1 MB boundary, of the prefetchable memory area of the bridge. Accesses from PCI Express* that are within the ranges specified in this register are sent to PCI when the Memory Space Enable bit is set. Accesses from PCI that are outside the ranges specified are forwarded to PCI Express* when the Bus Master Enable bit is set.

Note: Even though this register specifies a valid prefetchable memory window, the bridge never prefetches through this window in the downstream direction (reads from PCI Express*-to-PCI). Also, the bridge does not do any byte-merging in this window.

Note: Peer reads from PCI can prefetch through this window. This prefetching can be turned off with the prefetch policy bits (PP bits 42:41, "Offset 178h: PREFCTRL—Prefetch Control Register" on page 119).

These registers are cleared to all 0s on reset.

Note: This register must be programmed appropriately to enable or disable the space.

Table 48. Offset 24h: PMBL—Prefetchable Memory Base and Limit

Bits	Type	Reset	Description
31:20	RW	000h	**Prefetchable Memory Limit (PML):** These bits are compared with bits[31:20] of the incoming address to determine the upper 1 MB-aligned value (inclusive) of the range. The incoming address must be less than this value.
19:16	RO	1h	**64-bit Indicator (IS64L):** These bits indicate that 64-bit addressing is supported for the limit. This value must be in agreement with the IS64B field (bits[3:0], below).
15:4	RW	000h	**Prefetchable Memory Base (PMB):** These bits are compared with bits[31:20] of the incoming address to determine the lower 1 MB-aligned value (inclusive) of the range. The incoming address must be greater than or equal to this value.
3:0	RO	1h	**64-bit Indicator (IS64B):** These bits indicate that 64-bit addressing is supported for the limit. This value must be in agreement with the IS64L field (bits[19:16], above).

12.2.15 Offset 28h: PMBU32—Prefetchable Memory Base Upper 32 Bits

This register defines the upper 32 bits of the prefetchable address base register.

Table 49. Offset 28h: PMBU32—Prefetchable Memory Base Upper 32 Bits

Bits	Type	Reset	Description
31:0	RW	0000 0000h	**Prefetchable Memory Base Upper Portion (PMBU):** These bits indicate that full 64-bit addressing is supported.

12.2.16 Offset 2Ch: PMLU32—Prefetchable Memory Limit Upper 32 Bits

This register defines the upper 32 bits of the prefetchable address base register.

Table 50. Offset 2Ch: PMLU32—Prefetchable Memory Limit Upper 32 Bits

Bits	Type	Reset	Description
31:0	RW	0000 0000h	**Prefetchable Memory Limit Upper Portion (PMLU):** These bits indicate that full 64-bit addressing is supported.

12.2.17 Offset 30h: IOBLU16—I/O Base and Limit Upper 16 Bits

Since I/O is limited to 64 KB, this register is reserved and not used.

Table 51. Offset 30h: IOBLU16—I/O Base and Limit Upper 16 Bits

Bits	Type	Reset	Description
31:16	RO	0000h	**I/O Base High 16 Bits (IOBH):** Reserved
15:0	RO	0000h	**I/O Limit High 16 Bits (IOLH):** Reserved

12.2.18 Offset 34h: CAPP—Capabilities List Pointer

This register contains the pointer for the first entry in the capabilities list.

Table 52. Offset 34h: CAPP—Capabilities List Pointer

Bits	Type	Reset	Description
7:0	RO	44h	**Capabilities Pointer (PTR):** These bits indicate that the pointer for the first entry in the capabilities list is at 44h (PCI Express* capability) in the configuration space.

12.2.19 Offset 3Ch: INTR—Interrupt Information

This register contains information on interrupts on the bridge.

Table 53. Offset 3Ch: INTR—Interrupt Information

Bits	Type	Reset	Description
15:8	RO	00h	**Interrupt Pin (PIN):** These bits indicate that no interrupt is used by the bridge segment.
7:0	RW	00h	**Interrupt Line (LINE):** This register is used to convey the interrupt line routing information between the initialization code and the device driver.

12.2.20 Offset 3Eh: BCTRL—Bridge Control

This register provides extensions to the Command Register ("Offset 04h: PCICMD—Command Register" on page 78) that are specific to a bridge. The Bridge Control Register provides many of the same controls for the secondary interface that are provided by the Command Register for the primary interface. Some bits affect operation of both interfaces of the bridge.

Table 54. Offset 3Eh: BCTRL—Bridge Control (Sheet 1 of 2)

Bits	Type	Reset	Description
15:12	RO	0h	Reserved
11	RW	0b	**Discard Timer SERR# Enable (DTSE):** This bit controls the generation of ERR_NONFATAL/ERR_FATAL messages on the primary interface in response to a timer discard on the secondary interface. 0 = The 41210 does not generate ERR_NONFATAL/ERR_FATAL on a secondary timer discard. 1 = When the appropriate mask bit in the advanced capability register is clear, the 41210 does generate ERR_NONFATAL/ERR_FATAL in response to a secondary timer discard. **NOTE:** ERR_NONFATAL/ERR_FATAL messages may also be generated when the corresponding mask bit in the Uncorrectable Error Mask Register ("Offset 108h: ERRUNC_MSK—PCI Express* Uncorrectable Error Mask" on page 106) is cleared.
10	RWC	0b	**Discard Timer Status (DTS):** This bit is set to 1 when the secondary discard timer expires (there is no discard timer for the primary interface).
9	RW	0b	**Secondary Discard Timer (SDT):** This bit sets the maximum number of PCI clock cycles during which the bridge waits for an initiator on PCI to repeat a delayed transaction request. The counter starts as soon as the delayed transaction completion is at the head of the queue. When the master has not repeated the transaction at least once before the counter expires, the bridge discards the transaction from its queues. 0 = The PCI master time-out value is between 2^{15} and 2^{16} PCI clocks. 1 = The PCI master time-out value is between 2^{10} and 2^{11} PCI clocks.
8	RW	0b	**Primary Discard Timer (PDT):** This bit is not relevant to PCI Express*.
7	RO	0b	**Fast Back-to-Back Enable (FBE):** The bridge cannot generate fast back-to-back cycles on the PCI bus from PCI Express*-initiated transactions.
6	RW	0b	**Secondary Bus Reset (SBR):** This bit controls x_RST# assertion on PCI. 0 = The bridge does ***not*** force x_RST# assertion on the secondary interface. 1 = The bridge asserts PCIRST#. Bridge configuration registers are not reset when this bit is set. As soon as this bit is set, the bridge completes the currently running transaction on PCI and then resets the bus. Note that it is the responsibility of the software to ensure that all pending transactions with the bus segment are complete before setting this bit. When the software fails to do this, transactions can be lost. **NOTE:** Software must ensure the secondary bus x_RST# timing requirements when clearing this bit.

Table 54. Offset 3Eh: BCTRL—Bridge Control (Sheet 2 of 2)

Bits	Type	Reset	Description
5	RW	0b	**Master Abort Mode (MAM):** This bit controls the bridge's behavior when a master-abort (or unsupported request) occurs on either interface. This bit does not affect the behavior when the bridge forwards a UR completion from PCI Express* to master-abort on PCI-X. 0 = Do not report master-aborts. When a UR response is received from PCI Express* for non-posted transactions, and when the secondary side is operating in conventional PCI mode, the device returns FFFF FFFFh on reads and completes I/O writes normally. For posted transactions, the data is discarded and no additional action is taken. 1 = Report UR completions by signaling a target-abort on the secondary/peer interface when operating in conventional PCI mode. The device returns ERR_NONFATAL/ERR_FATAL messages for posted transactions initiated from PCI Express*.
4	RW	0b	**VGA 16-bit Decode:** This bit enables the bridge to provide 16-bit decoding of VGA I/O address precluding the decoding of VGA alias addresses every 1 KB. This bit requires the VGA enable bit (bit 3 of this register) to be set to 1.
3	RW	0b	**VGA Enable (VGAE):** This bit modifies the response to VGA-compatible addresses. When set to 1, the bridge forwards the following transactions from PCI Express*-to-PCI regardless of the value of the I/O base and limit registers. The transactions are qualified by the memory enable and I/O enable in the command register. **Memory addresses:** 000A 0000h–000B FFFFh **I/O addresses:** 3B0h–3BBh and 3C0h–3DFh. For the I/O addresses, bits[63:16] of the address must be 0, and bits[15:10] of the address are ignored (aliased). The same holds true from secondary accesses to the primary interface in reverse for memory accesses and also for I/O when the upstream I/O enable bit is set in the BINIT register, from secondary to primary.
2	RW	0b	**ISA Enable (IE):** This bit modifies the response by the bridge to ISA I/O addresses. This function applies only to I/O addresses that are enabled by the I/O base and I/O limit registers and are in the first 64 KB of PCI I/O space. When this bit is set, the bridge blocks all forwarding from primary to secondary of I/O transactions addressing the last 768 bytes in each 1 KB block (offsets 100h to 3FFh). This bit has the reverse effect on I/O transfers originating on the secondary bus when the upstream I/O enable bit is set in the BINIT Register ("Offset FCh: BINIT—Bridge Initialization Register" on page 104).
1	RW	0b	**SERR# Enable (SE):** This bit controls the forwarding of secondary interface SERR# assertions on the primary interface. When set, the bridge sends a PCI Express* ERR_NONFATAL/ERR_FATAL cycle (based on Advanced Error capability's PCI SERR detected severity bit) when all of the following conditions are true: • SERR# is asserted on the secondary interface. • This bit is set. • The SERR# detected mask bit in the Advanced Error capability is set. • ERR_NONFATAL/ERR_FATAL messages are enabled to be sent.
0	RW	0b	**Parity Error Response Enable (PERE):** This bit controls the response to address and data parity errors on the secondary interface. When the bit is cleared, the bridge must ignore any parity errors that it detects and continue normal operation. The bridge must generate parity even when parity error reporting is disabled.

12.2.21 Offset 40h: BCNF—Bridge Configuration Register

The bridge control bits specific to the Intel® 41210 Serial to Parallel PCI Bridge are listed in Table 55.

Table 55. Offset 40h: BCNF—Bridge Configuration Register

<table>
<tr><th>Bits</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>15</td><td>RsvdP</td><td>0h</td><td>Preserved</td></tr>
<tr><td>14</td><td>RW</td><td>See Table 11 on page 25.</td><td>PCI Mode (PMODE): This bit determines the mode of operation of the PCI bus. This bit reflects the status of the current PCI bus mode and also allows the software to change the mode by writing to this bit. The power-up value of this register is written based upon Table 11, “PCI Mode Pin/Strap Encoding” on page 25.
0 = Conventional PCI
1 = PCI-X</td></tr>
<tr><td>13:11</td><td>RsvdP</td><td>101b</td><td>Preserved</td></tr>
<tr><td>10:9</td><td>RW</td><td>See Table 12 on page 25.</td><td>PCI Frequency (PFREQ): This bit determines the frequency at which the PCI bus operates. The power-up value of this register is written based upon Table 12, “PCI-X Initialization Pattern” on page 25. After the software determines the capabilities of the bus, it sets this value, and the PMODE bit (bit[14] of this register) to the desired frequency and resets the PCI bus. The values are encoded as follows:
<table>
<tr><th>Bit[10:9]</th><th>Frequency</th><th>Comments</th></tr>
<tr><td>00</td><td>33 MHz</td><td>Valid only when PMODE is 0.</td></tr>
<tr><td>01</td><td>66 MHz</td><td></td></tr>
<tr><td>10</td><td>100 MHz</td><td>Valid only when PMODE is 1.</td></tr>
<tr><td>11</td><td>133 MHz</td><td>Valid only when PMODE is 1.</td></tr>
</table>
Results are indeterminate when invalid combinations are written by the software.</td></tr>
<tr><td>8</td><td>RsvdP</td><td>0b</td><td>Preserved</td></tr>
<tr><td>7</td><td>RW</td><td>1b</td><td>Peer Memory Read Enable (PMRE):
0 = Normal operation. Peer memory reads are not allowed. All memory reads from a PCI bridge are sent to PCI Express* regardless of address.
1 = Peer memory reads from one PCI segment to another are supported.</td></tr>
<tr><td>6:4</td><td>RsvdP</td><td>000b</td><td>Preserved</td></tr>
<tr><td>3</td><td>RsvdP</td><td>0b</td><td>Preserved</td></tr>
<tr><td>2</td><td>RsvdP</td><td>0b</td><td>Preserved</td></tr>
<tr><td>1:0</td><td>RW</td><td>00b</td><td>Maximum Upstream Delayed Transactions (MDT): This bit controls the maximum number of upstream delayed transactions the Intel® 41210 Serial to Parallel PCI Bridge is allowed to have:
<table>
<tr><th>Bit</th><th>Maximum Number of Upstream Delayed Transactions</th></tr>
<tr><td>00</td><td>4 active, 4 pending</td></tr>
<tr><td>01</td><td>1 active, 1 pending</td></tr>
<tr><td>10</td><td>2 active, 2 pending</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table></td></tr>
</table>

12.2.22 Offset 42h: MTT—Multi-Transaction Timer

This register controls the amount of time that the 41210 arbiter allows for a PCI initiator to perform multiple back-to-back transactions on the PCI bus. The number of clocks programmed in the MTT represents the time slice (measured in PCI clocks) to be allotted to the current agent, after which the arbiter grants the bus to another agent that is requesting it.

Table 56. Offset 42h: MTT—Multi-Transaction Timer

Bits	Type	Reset	Description
7:3	RW	00h	**Timer Count Value (MTC):** This field specifies the amount of time that the grant remains asserted to a master that is continuously asserting its request for multiple transfers. This field specifies the count in an 8-clock (PCI clock) granularity.
2:0	RsvdP	000b	Preserved

12.2.23 Offset 43h: PCLKC—PCI Clock Control

This register controls the enable and disable of the 41210 PCI clock outputs.

Table 57. Offset 43h: PCLKC—PCI Clock Control

Bits	Type	Reset	Description
7	RsvdP	1b	Reserved
6	RW	1b	**PCI Feedback Clock Control:** 0 = PCI feedback clock output buffer, X_CLK[6], is tristated. 1 = PCI feedback clock output buffer, X_CLK[6], is enabled.
5:0	RW	1 1111b	**PCI Clock Control:** These bits enable the PCI clock output buffers, when all 1s. Otherwise the buffers are tristated. Bit[0] corresponds to X_CLKO[0], bit[1] corresponds to X_CLKO[1], etc. The tristating of the clock is asynchronous to the output clocks.

12.2.24 Offset 44h: EXP_CAPID—PCI Express* Capability Identifier

This register stores the PCI Express* capability ID value.

Table 58. Offset 44h: PCI Express*_CAPID—PCI Express* Capability Identifier

Bits	Type	Reset	Description
7:0	RO	10h	**PCI Express* Capability ID:** These bits indicate that the Intel® 41210 Serial to Parallel PCI Bridge has PCI Express* capability.

12.2.25 Offset 45h: EXP_NXTP—Next Item Pointer

This register stores the byte offset pointer to the next capability list item of the bridge.

Table 59. Offset 45h: PCI Express*_NXTP—Next Item Pointer

Bits	Type	Reset	Description
7:0	RO	5Ch	**Next Capability Pointer:** This field indicates the offset of the next capabilities list item, which is the MSI capability.

12.2.26 Offset 46h: EXP_CAP—PCI Express* Capability

This register stores the version number of the capability item and other base information contained in the capability structure.

Table 60. Offset 46h: EXP_CAP—PCI Express* Capability

Bits	Type	Default	Description
15:14	RsvdP	00b	Preserved
13:9	RO	0 0000b	**Interrupt Message Number:** Not relevant for the Intel® 41210 Serial to Parallel PCI Bridge
8	RO	0b	**Slot Implemented:** Not relevant for the 41210
7:4	RO	7h	**Device/Port Type:** These bits indicate that the 41210 is a PCI Express* end-point device.
3:0	RO	1h	**Version Number:** These bits indicate the version number of the PCI Express* capability structure.

12.2.27 Offset 48h: EXP_DCAP—PCI Express* Device Capabilities Register

This register stores information on the PCI Express* link capabilities.

Table 61. Offset 48h: EXP_DCAP—PCI Express* Device Capabilities Register

Bits	Type	Default	Description
31:28	RsvdP	0h	Preserved
27:26	RO	00b	**Captured Slot Power Limit Scale:** In combination with the Slot Power Limit value (bits[25:18], this field specifies the upper limit of the power supplied by slot. The power limit (in Watts) is calculated by multiplying the value in this field by the value in the Slot Power Limit Value field. This value is set by the Set_Slot_Power_Limit message.
25:18	RO	00h	**Captured Slot Power Limit Value:** In combination with the Slot Power Limit Scale value (bits[27:26]), this field specifies the upper limit of the power supplied by slot. The power limit (in Watts) is calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field. This value is set by the Set_Slot_Power_Limit message.
17:15	RsvdP	000b	Preserved
14	RO	0b	**Power Indicator Present:** Not supported
13	RO	0b	**Attention Indicator Present:** Not supported
12	RO	0b	**Attention Button Present:** Not supported
11:9	RO	000b	**Endpoint L1 Acceptable Latency:** L1 ASPM is not supported.
8:6	RO	000b	**Endpoint L0s Acceptable Latency:** The least latency possible out of L0s is supported. **NOTE:** L0s ASPM is not supported in the 41210 Bridge.
5	RO	0b	**Extended Tag Field Supported:** Only a 5-bit tag is supported.
4:3	RO	00b	**Phantom Functions Supported:** Not supported
2:0	RO	001b	**Supported Max Payload sizes:** 256-byte packets are the maximum supported.

12.2.28 Offset 4Ch: EXP_DCTL—PCI Express* Device Control Register

This register stores command bits that control the 41210 behavior on PCI Express*.

Table 62. Offset 4Ch: EXP_DCTL—PCI Express* Device Control Register (Sheet 1 of 2)

Bits	Type	Default	Description
15	RW	0b	**Bridge Configuration Retry Enable:** When set, the bridge is enabled to return a completion with Completion Retry Status (CRS) on PCI Express* when a configuration transaction to the secondary interface did not complete within the PCI completion time-out period.
14:12	RW	010b	**Max_Read_Request_Size:** This field applies to the bridge segment when the segment is in PCI mode only. When in PCI-X mode, this field does not apply. The Intel® 41210 Serial to Parallel PCI Bridge cannot send requests larger than the size indicated by this field. Encodings are as follows: Bit: Max_Read_Request_Size 000b: 128-byte maximum read request size 001b: 256-byte maximum read request size 010b: 512-byte maximum read request size 011b: 1024-byte maximum read request size 100b: 2048-byte maximum read request size 101b: 4096-byte maximum read request size 110b: Reserved (the 41210 defaults to 512 bytes) 111b: Reserved (the 41210 defaults to 512 bytes)
11	RO	0b	**Enable No Snoop:** Hard-wired to 0
10	RO	0b	**Auxiliary (AUX) Power PM Enable:** Not supported
9	RO	0b	**Phantom Function Enable:** Not supported
8	RO	0b	**Extended Tag Field Enable:** Ignored because only 5-bit tag is supported
7:5	RW	000b	**Maximum Payload Size:** These bits indicate the maximum payload size supported for TLPs. Supported encodings are as follows: Bit: Max_Payload_Size 000b: 128-byte maximum payload size 001b: 256-byte maximum payload size All other values default to 128 bytes.
4	RO	0b	**Enable Relaxed Ordering:** Hard-wired to 0
3	RW	0b	**Unsupported Request Reporting Enable:** This bit controls the enabling of ERR_NONFATAL or ERR_FATAL messages on PCI Express* for reporting “Unsupported Request” errors. Note that the following requests use this enable bit: • requests from PCI Express* that are unsupported • requests from PCI Express* that master-abort on the internal switch

Table 62. Offset 4Ch: EXP_DCTL—PCI Express* Device Control Register (Sheet 2 of 2)

Bits	Type	Default	Description
2	RW	0b	**Report Fatal Errors:** When this bit is set, generation of the ERR_FATAL message is enabled.
1	RW	0b	**Report NonFatal Errors:** When this bit is set, generation of the ERR_NONFATAL message is enabled.
0	RW	0b	**Report Correctable Errors:** When this bit is set, generation of the ERR_CORR message is enabled.

12.2.29 Offset 4Eh: EXP_DSTS—PCI Express* Device Status Register

This register stores information on the PCI Express* device status.

Table 63. Offset 4Eh: EXP_DSTS—PCI Express* Device Status Register

Bits	Type	Default	Description
15:6	RsvdZ	000h	**Reserved Zero:** Software must always write a 0 to these bits.
5	RO	0b	**Transactions Pending:** This bit is set when any non-posted request has been issued but has not been completed. The bit is cleared only when all completions for all outstanding non-posted requests are received. Note that this is a dynamic bit; in other words, this bit goes on and off based on current traffic.
4	RO	0b	**Auxiliary Power Detected:** Auxiliary Power is not supported.
3	RWC	0b	**Unsupported Request Detected:** This bit is set when any unsupported request from PCI Express* is received. Unsupported requests include all requests that are not claimed by any function in the Intel® 41210 Serial to Parallel PCI Bridge, but do ***not*** include requests forwarded to the PCI interface with completions returned with an “unsupported request” status.
2	RWC	0b	**Detected Fatal Error:** This bit is set when a fatal error is detected (regardless of whether an error message is generated) on either interface or internally. The bit remains set until the software writes a 1 to clear it.
1	RWC	0b	**Detected Non-Fatal Error:** This bit is set when a non-fatal error is detected (regardless of whether the mask bit is set in the advanced error capability) on either interface or internally. The bit remains set until the software writes a 1 to clear it.
0	RWC	0b	**Detected Correctable Error:** This bit is set when a correctable error is detected (regardless of whether the mask bit is set in the advanced error capability) on either interface or internally. The bit remains set until the software writes a 1 to clear it.

12.2.30 Offset 50h: EXP_LCAP—PCI Express* Link Capabilities Register

Table 64. Offset 50h: EXP_LCAP—PCI Express* Link Capabilities Register (Sheet 1 of 2)

Bits	Type	Default	Description
31:24	RO	00h	**Port Number:** Not applicable
23:18	RsvdP	00h	Preserved
17:15	RO	111b	**L1 Exit Latency:** L1 transition is not supported.

Table 64. Offset 50h: EXP_LCAP—PCI Express* Link Capabilities Register (Sheet 2 of 2)

Bits	Type	Default	Description
14:12	RO	110b	**L0s Exit Latency:** The value in these bits is determined by the setting of the Common Clock Configuration bit (bit[6]) in the Link Control Register (Offset 54h: EXP_LCTL—PCI Express* Link Control Register). Note that software can write bit[6] in the Link Control Register to either a 1 or 0 and these bits then change accordingly. The mapping is shown below: **Bit 6 in LCTL** / **L0s Exit Latency** 0 / 110b (because currently L0s cannot work with different reference clocks) 1 / 010b **NOTE:** L0s ASPM is not supported in the 41210 Bridge.
11:10	RO	01b	L0s ASPM is not supported in the 41210 Bridge.
9:4	RO	08h	**Maximum Link Width:** X8 link width is supported.
3:0	RO	1h	**Maximum Link Speed:** 2.5 Gb/s link speed is supported.

12.2.31 Offset 54h: EXP_LCTL—PCI Express* Link Control Register

Table 65. Offset 54h: EXP_LCTL—PCI Express* Link Control Register

Bits	Type	Default	Description
15:8	RsvdP	00h	Preserved
7	RW	0b	**Extended Synch.:** When set, this bit forces extended transmission of 4096 FTS ordered sets in FTS and an extra 1024 TS1 at exit from L1 prior to entering L0. This mode provides external devices monitoring the link time to achieve bit and symbol lock before the link enters L0 state and resumes communication. Default value for this bit is 0.
6	RW	0b	**Common Clock Configuration:** This bit indicates the relationship of the reference clock between the Intel® 41210 Serial to Parallel PCI Bridge and the component at the opposite end of the 41210 Upstream PCI Express* interface: 0 = Clock is asynchronous. 1 = Clock is common. **NOTE:** This bit determines the proper L0s exit latency value in the EXP_LSTS register. **NOTE:** L0s ASPM is not supported in the 41210 Bridge.
5	RO	0b	**Retrain Link:** Not applicable
4	RO	0b	**Disable Link:** Not applicable
3	RO	0b	**Read Completion Boundary Control:** Not used
2	RsvdP	0b	Preserved
1:0	RW	00b	**ASPM Control:** Enables bridge upstream interface to enter L0s: • 00b = L0s entry is disabled. • 01b = The Intel® 41210 Serial to Parallel PCI Bridge enters L0s as per the specification requirement for L0s entry. • 10b = L0s entry is disabled. • 11b = The 41210 enters L0s as per the specification requirement for L0s entry. **NOTE:** L0s ASPM is not supported in the 41210 Bridge.

12.2.32 Offset 56h: EXP_LSTS—PCI Express* Link Status Register

Table 66. Offset 56h: EXP_LSTS—PCI Express* Link Status Register

Bits	Type	Default	Description
15:13	RsvdZ	000b	**Reserved Zero:** Software must always write 0 to these bits.
12	RO	1b	**Slot Clock Configuration:** When the Intel® 41210 Serial to Parallel PCI Bridge is on a PCI Express* connector, this bit indicates whether it is using the same reference clock that is provided at the connector. 0 = Indicates independent reference clock 1 = Indicates same reference clock.
11	RO	0b	**Link Training:** Not applicable
10	RO	0b	**Link Width Negotiation Error:** Not applicable
9:4	RO	Set by PCI Express* Link Layer after training is complete	**Negotiated Link Width:** This field indicates the negotiated width of the PCI Express* link. • 00 0001b X1 • 00 0010b X2—not supported • 00 0100b X4 • 00 1000b X8 • 00 1100b X12—not supported • 01 0000b X16—not supported • 10 0000b X32—not supported • All other values are reserved.
3:0	RO	1h	**Link Speed:** The only speed supported is 2.5 Gbps.

12.2.33 Offset 5Ch: MSI_CAPID—PCI Express* MSI Capability Identifier

Note: MSI generation is used for internal debugging purposes and does not occur in normal operation.

Table 67. Offset 5Ch: MSI_CAPID—PCI Express* MSI Capability Identifier

Bits	Type	Reset	Description
7:0	RO	05h	**Capability ID (MCID):** Capabilities ID indicates MSI.

12.2.34 Offset 5Dh: MSI_NXTP—PCI Express* Next Item Pointer

Table 68. Offset 5Dh: MSI_NXTP—PCI Express* Next Item Pointer

Bits	Type	Reset	Description
7:0	RO	6Ch	**Next Pointer (MNPTR):** This field points to the next capabilities list pointer, which is the PCI Express* Power Management capability item.

12.2.35 Offset 5Eh: MSI_MC—PCI Express* MSI Message Control

Table 69. Offset 5Eh: MSI_MC—PCI Express* MSI Message Control

Bits	Type	Reset	Description
15:8	RO	00h	Reserved
7	RO	1b	**64-Bit Address Capable:** When set, this bit indicates that the Intel® 41210 Serial to Parallel PCI Bridge is capable of generating a 64-bit message address. (Set by default.)
6:4	RW	000b	**Multiple Message Enable:** Only one message is supported. These bits are R/W for software compatibility.
3:1	RO	000b	**Multiple Message Capable:** Only one message is supported.
0	RW	0b	**MSI Enable:** When set, MSI is enabled and traditional interrupt pins are not used to generate interrupts.

12.2.36 Offset 60h: MSI_MA—PCI Express* MSI Message Address

Table 70. Offset 60h: MSI_MA—PCI Express* MSI Message Address

Bits	Type	Reset	Description
63:2	RW	0	**Address (ADDR):** Message address specified by the system, always DWORD aligned
1:0	RO	00b	Reserved

12.2.37 Offset 68h: MSI_MD—PCI Express* MSI Message Data

Table 71. Offset 68h: MSI_MD—PCI Express* MSI Message Data

Bits	Type	Reset	Description
15:0	RW	0000h	**Data (DATA):** This 16-bit field is programmed by system software when MSI is enabled. Its content is driven onto the lower word (D[15:0]) of the MSI memory write transaction.

12.2.38 Offset 6Ch: PM_CAPID—Power Management Capabilities Identifier

Table 72. Offset 6Ch: PM_CAPID—Power Management Capabilities Identifier

Bits	Type	Reset	Description
7:0	RO	01h	**Identifier (ID):** These bits indicate that this is a PCI-compatible PM.

12.2.39 Offset 6Dh: PM_NXTP—Power Management Next Item Pointer

Table 73. Offset 6Dh: PM_NXTP—Power Management Next Item Pointer

Bits	Type	Reset	Description
7:0	RO	D8h	**Next Pointer:** This field points to the PCI-X capability as the next capability.

12.2.40 Offset 6Eh: PM_PMC—Power Management Capabilities

Table 74. Offset 6Eh: PM_PMC—Power Management Capabilities

Bits	Type	Reset	Description
15:11	RO	19h	**PME_Support:** PME assertion is supported when in D3hot. PME assertion from D3cold is not supported.
10	RO	0b	**D2 Support:** Not supported
9	RO	0b	**D1 Support:** Not supported
8:6	RO	000b	**Auxiliary Current:** Auxiliary power is not supported.
5	RO	0b	**DSI:** Device-specific initialization is not required when transitioning to D0 from D3hot state. This bit is zero.
4	RsvdP	0b	Preserved
3	RO	0b	**PME Clock:** Does not apply to PCI Express*. Hard-wired to 0.
2:0	RO	2h	**Version:** PM implementation is compliant with *PCI Bus Power Management Interface Specification*, Revision 1.1.

12.2.41 Offset 70h: PM_PMCSR—Power Management Control/Status Register

Table 75. Offset 70h: PM_PMCSR—Power Management Control/Status Register

Bits	Type	Reset	Description
15	RO	0b	**PME Status:** Not supported
14:13	RO	00h	**Data Scale:** Not supported
12:9	RO	0h	**Data Select:** Not supported
8	RWS	0b	**PME En:** Not supported
7:2	RsvdP	00 0000b	Preserved
1:0	RW	00b	**Power State:** This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state. Supported field values are given below. • 00b = D0 • 01b = Reserved • 10b = Reserved • 11b = D3 hot When the software attempts to write an unsupported, optional state to this field, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs.

12.2.42 Offset 72h: PM_BSE—Power Management Bridge Support Extensions

Table 76. Offset 72h: PM_BSE—Power Management Bridge Support Extensions

Bits	Type	Reset	Description
7	RO	0b	**BPCC_En (Bus Power/Clock Control Enable):** Neither bus or clock control of PCI is supported when in D3hot state. This bit is hard-wired to 0.
6	RO	0b	**B2/B3#:** Not supported. This bit has no meaning since the BPCC_En bit is hard-wired to 0.
5:0	RsvdP	00h	Preserved

12.2.43 Offset 73h: PM_DATA—Power Management Data Field

Table 77. Offset 73h: PM_DATA—Power Management Data Field

Bits	Type	Reset	Description
7:0	RO	00h	**Data:** Not supported

12.2.44 Offset D8h: PX_CAPID—PCI-X Capabilities Identifier

This register identifies this item in the capabilities list as a PCI-X register set.

Table 78. Offset D8h: PX_CAPID—PCI-X Capabilities Identifier

Bits	Type	Reset	Description
7:0	RO	07h	**Identifier (ID):** These bits indicate that this is a PCI-X capabilities list.

12.2.45 Offset D9h: PX_NXTP—PCI-X Next Item Pointer

This register indicates where the next item in the capabilities list resides. This is the end of the list, so “00H” is returned.

Table 79. Offset D9h: PX_NXTP—PCI-X Next Item Pointer

Bits	Type	Reset	Description
7:0	RO	00h	PCI-X is the last capability list item and hence these bits are all 0s.

12.2.46 Offset DAh: PX_SSTS—PCI-X Secondary Status

This is the PCI-X status register for the bridge secondary side.

Table 80. Offset DAh: PX_SSTS—PCI-X Secondary Status

<table>
<tr><th>Bits</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>15:9</td><td>RO</td><td>00h</td><td>Reserved</td></tr>
<tr><td>8:6</td><td>RO</td><td>See Table 12 on page 25</td><td>Secondary Clock Frequency (SCF): This field is set with the frequency of the secondary bus. The values are as follows:
<table><tr><th>Bits</th><th>Maximum Frequency</th><th>Clock Period</th></tr><tr><td>000</td><td>PCI mode</td><td>–</td></tr><tr><td>001</td><td>66 MHz</td><td>15</td></tr><tr><td>010</td><td>100 MHz</td><td>10</td></tr><tr><td>011</td><td>133 MHz</td><td>7.5</td></tr><tr><td>1xx</td><td>Reserved</td><td>Reserved</td></tr></table>The default value for this register is given in Table 12, “PCI-X Initialization Pattern” on page 25.</td></tr>
<tr><td>5</td><td>RO</td><td>0b</td><td>Split Request Delayed. (SRD): Ordinarily, this bit is set when the bridge cannot forward a transaction on the secondary bus to the primary bus because there is not enough room within the limit specified in the Split Transaction Commitment Limit field in the Downstream Split Transaction Control Register.
The Intel® 41210 Serial to Parallel PCI Bridge never sets this bit.</td></tr>
<tr><td>4</td><td>RO</td><td>0b</td><td>Split Completion Overrun (SCO): Ordinarily, this bit is set when a bridge terminates a Split Completion on the secondary bus with retry or disconnect at next ADB because its buffers are full.
The 41210 never sets this bit.</td></tr>
<tr><td>3</td><td>RWC</td><td>0b</td><td>Unexpected Split Completion (USC): This bit is set when an unexpected split completion is received with a bus number that matches the 41210 primary bus number but with a requester ID:tag that does not match any outstanding requests. This bit is cleared by the software writing a 1.</td></tr>
<tr><td>2</td><td>RWC</td><td>0b</td><td>Split Completion Discarded (SCD): This bit is set when the 41210 discards a split completion moving toward the secondary bus because the requester does not accept it. This bit is cleared by the software writing a 1.</td></tr>
<tr><td>1</td><td>RO</td><td>1b</td><td>133 MHz Capable (C133): This bit indicates that the 41210 secondary interface is capable of 133 MHz operation in PCI-X mode.</td></tr>
<tr><td>0</td><td>RO</td><td>1b</td><td>64-bit Device (D64): This bit indicates that the width of the secondary bus is 64 bits.</td></tr>
</table>

12.2.47 Offset DCh: PX_BSTS—PCI-X Bridge Status

This register identifies PCI-X status register for the bridge primary side.

Table 81. Offset DCh: PX_BSTS—PCI-X Bridge Status

Bits	Type	Reset	Description
31:22	RO	000h	Reserved
21	RO	0b	**Split Request Delayed (SRD):** Ordinarily, this bit is set by a bridge when it cannot forward a transaction onto the primary bus from the secondary bus because there is not enough room within the limit specified in the Split Transaction Commitment Limit field in the upstream Split Transaction Control Register. The Intel® 41210 Serial to Parallel PCI Bridge does not set this bit.
20	RO	0b	**Split Completion Overrun (SCO):** This bit is not set by the 41210 because the 41210 does not request more data on PCI Express* than it has room to receive.
19	RO	0b	**Unexpected Split Completion (USC):** This bit is set when a completion on PCI Express* is addressed to a specific bridge segment (either A or B) but the tag does not match.
18	RO	0b	**Split Completion Discarded (SCD):** This bit does not apply to PCI Express*.
17	RO	0b	**133 MHz Capable (C133):** Not applicable
16	RO	0b	**64-bit Device (D64):** Not applicable
15:8	RO	00h	**Bus Number (BNUM):** This register is an alias to the PBN field of the BNUM register ("Offset 18h: BNUM—Bus Numbers" on page 82).
7:3	RO	00h	**Device Number (DNUM):** Device number is 0 for both PCI segment bridges.
2:0	RO	A: 000b / B: 010b	**Function Number (FNUM):** Read-only bits for PCI-X diagnostic software

12.2.48 Offset E0h: PX_USTC—PCI-X Upstream Split Transaction Control

This register controls the behavior of the 41210 buffers for forwarding split transactions from the secondary bus to PCI Express*.

Table 82. Offset E0h: PX_USTC—PCI-X Upstream Split Transaction Control

Bits	Type	Reset	Description
31:16	RW	FFFFh	**Split Transaction Limit (STL):** This field is R/W to accommodate diagnostic software that might want to use it. This field is not used by the Intel® 41210 Serial to Parallel PCI Bridge for modifying its "commitment" level. The 41210 internal launch algorithms prevent buffers from being over-allocated.
15:0	RO	FFFFh	**Split Transaction Capacity (STC):** Due to the internal launch algorithm, the 41210 always has capacity for its outstanding split transactions.

12.2.49 Offset E4h: PX_DSTC—PCI-X Downstream Split Transaction Control

This register controls the behavior of the 41210 buffers for forwarding split transactions from PCI Express* to the secondary bus.

Table 83. Offset E4h: PX_DSTC—PCI-X Downstream Split Transaction Control

Bits	Type	Reset	Description
31:16	RW	FFFFh	**Split Transaction Limit (STL):** This field is R/W to accommodate diagnostic software that might want to use it. The field is not used by the Intel® 41210 Serial to Parallel PCI Bridge for modifying its “commitment” level. The 41210 internal launch algorithms prevent buffers from being over-allocated.
15:0	RO	FFFFh	**Split Transaction Capacity (STC):** Due to the internal launch algorithm, the 41210 always has capacity for its outstanding split transactions.

12.2.50 Offset FCh: BINIT—Bridge Initialization Register

Table 84. Offset FCh: BINIT—Bridge Initialization Register

Bits	Type	Reset	Description
31:5	RO	000 0000h	Reserved
4	RWS	0b	**Opaque Memory Window Enable:** When this bit is set, the Intel® 41210 Serial to Parallel PCI Bridge hard-codes certain address ranges to the *secondary* segment of each bridge. The hard-coded address ranges are as follows: • A[63:62] = 10 secondary side of A-segment • A[63:63] = 11 secondary side of B-segment This address range is forwarded from the primary to the corresponding secondary side and also is never forwarded from the secondary to the primary side, irrespective of the setting of the prefetchable base and limit registers. Note that even when the opaque memory window is enabled, the normal 41210 behavior defined for the BME, MSE, and IOSE bits in the PCICMD register are still true ("Offset 04h: PCICMD—Command Register" on page 78). 0 = Opaque Memory Window disabled 1 = Opaque Memory Window enabled Each segment of the bridge uses the bit corresponding to that side.
3	RW	Varies with external state of **CFGRETRY** pin on rising edge of **PERST#**	**Configuration Cycle Retry:** When set, this bit results in a configuration retry response on the Intel® 41210 Serial to Parallel PCI Bridge for all Type 0 configuration transactions from PCI Express* to the internal bridge registers. When this bit is cleared, Type 0 transactions are completed normally to the internal 41210 bridge registers. 0 = Normal response to Type 0 configuration transactions 1 = Response to Type 0 configuration transaction is the Configuration Retry Response. Each segment of the bridge uses the bit corresponding to that side.
2	RWS	0b	**Device Hiding Enable:** This bit enables hiding of devices on the secondary PCI bus: 0 = All downstream devices are visible to all requestors. 1 = Device numbers 0 to 9 on the immediate secondary bus are hidden from downstream configuration transactions. See Section 5.2, "Secondary PCI Devices" on page 42. Each segment of the bridge uses the bit corresponding to that side.
1	RWS	1b	**Upstream Configuration Enable:** This bit controls the behavior for upstream configuration transactions allowing local initialization from the secondary PCI-X interface: 0 = Upstream configuration transactions are not claimed. 1 = Upstream configuration transactions with AD[16] = 1 are claimed. Each segment of the bridge uses the bit corresponding to that side.
0	RWS	1b	**Upstream I/O Enable:** When set, this bit enables I/O transactions from PCI. The I/O transactions are enabled by default.

12.2.51 Offset 100h: EXPAERR_CAPID—PCI Express* Advanced Error Capability Identifier

This register stores the PCI Express* extended capability ID value.

Table 85. Offset 100h: EXPAERR_CAPID—PCI Express* Advanced Error Capability Identifier

Bits	Type	Reset	Description
31:20	RO	300h	**Next PCI Express* Extended Capability Pointer:** This field points to the PCI Express* Power Budgeting Capability as the next capability.
19:16	RO	1h	**Advanced Error Capability Version Number:** This field indicates the PCI Express* Advanced Error Reporting Extended Capability Version Number.
15:0	RO	0001h	**Advanced Error Capability ID:** This field indicates the PCI Express* Extended Capability ID, indicating Advanced Error Reporting Capability.

12.2.52 Offset 104h: ERRUNC_STS—PCI Express* Uncorrectable Error Status Register

This register reports the error status of individual uncorrectable error sources. An individual error status bit that is set to 1 indicates that a particular error occurred. Software can clear an error status by writing a 1 to the respective bit.

Table 86. Offset 104h: ERRUNC_STS—PCI Express* Uncorrectable Error Status Register

Bits	Type	Reset	Description
31:21	RsvdZ	000h	**Reserved Zero:** Software must write 0 to these bits.
20	RWCS	0b	**Unsupported Request Error Status:** This bit is set whenever an unsupported request is detected on PCI Express*.
19	RO	0b	**ECRC Check:** The Intel® 41210 Serial to Parallel PCI Bridge does not do ECRC checking, and this bit is never set.
18	RWCS	0b	**Malformed TLP:** This bit is set when it receives a malformed TLP. Header logging is performed.
17	RWCS	0b	**Receiver Overflow:** This bit is set when the PCI Express* interface unit receive buffers overflow.
16	RWCS	0b	**Unexpected Completion:** This bit is set whenever a completion is received with a requestor ID that does not match side A or side B, or when a completion is received with a matching requestor ID but an unexpected tag field. Header logging is performed.
15	RWCS	0b	**Completer Abort:** The bridge sets this bit and logs the header associated with the request when the configuration unit signals a completer abort.
14	RWCS	0b	**Completion Time-out:** This bit is set when upstream memory configuration I/O reads do not receive completions within 16–32 ms.
13	RWCS	0b	**Flow Control Protocol Error Status:** This bit is set when a flow control protocol error is detected.
12	RWCS	0b	**Poisoned TLP Received:** This bit is set and the bridge logs the header when a poisoned TLP is received from PCI Express*.
11:5	RsvdZ	00h	**Reserved Zero:** Software must write 0 to these bits.
4	RWCS	0b	**Data Link Protocol Error:** This bit is set when a data link protocol error is detected.
3:1	RsvdZ	000b	Reserved
0	RWCS	0b	**Training Error:** The 41210 does not set this bit.

12.2.53 Offset 108h: ERRUNC_MSK—PCI Express* Uncorrectable Error Mask

This register controls the reporting of individual uncorrectable errors by device to the host bridge via a PCI Express* error message. This register also controls the logging of the header. Refer to the PCI Express* specifications for details of how the mask bits function. A masked error (respective bit set in the mask register) is not reported to the host bridge by the 41210, nor is the header logged (status bits unaffected by the mask bit). There is a mask bit per bit of the Uncorrectable Error Status Register ("Offset 104h: ERRUNC_STS—PCI Express* Uncorrectable Error Status Register" on page 105).

Table 87. Offset 108h: ERRUNC_MSK—PCI Express* Uncorrectable Error Mask

Bits	Type	Reset	Description
31:21	RsvdP	000h	Preserved
20	RWCS	0b	**Unsupported Request Error Status Error Mask:** 0 = Not masked 1 = Masked
19	RO	0b	**ECRC Check Error Mask:** Not supported
18	RWCS	0b	**Malformed TLP Error Mask:** 0 = Not masked 1 = Masked
17	RWCS	0b	**Receiver Overflow Error Mask:** 0 = Not masked 1 = Masked
16	RWCS	0b	**Unexpected Completion Error Mask:** 0 = Not masked 1 = Masked
15	RWCS	0b	**Completer Abort Error Mask:** 0 = Not masked 1 = Masked
14	RWCS	0b	**Completion Time Out Error Mask:** 0 = Not masked 1 = Masked
13	RWCS	0b	**Flow Control Protocol Error Status Error Mask:** 0 = Not masked 1 = Masked
12	RWCS	0b	**Poisoned TLP Received Error Mask:** 0 = Not masked 1 = Masked
11:5	RsvdP	00h	Preserved
4	RWCS	0b	**Data Link Protocol Error Mask:** 0 = Not masked 1 = Masked
3:1	RsvdP	000b	Preserved
0	RO	0b	**Training Error Mask:** Not supported

12.2.54 Offset 10Ch: ERRUNC_SEV—PCI Express* Uncorrectable Error Severity

This register controls whether an individual uncorrectable error is reported as a fatal error. An uncorrectable error is reported as fatal when the corresponding error bit in this register is set. When the bit is cleared, the corresponding error is considered non-fatal.

Table 88. Offset 10Ch: ERRUNC_SEV—PCI Express* Uncorrectable Error Severity

Bits	Type	Reset	Description
31:21	RsvdP	000h	Preserved
20	RWCS	0b	**Unsupported Request Error Status Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
19	RO	0b	**ECRC Check Severity:** Not supported
18	RWCS	0b	**Malformed TLP Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
17	RWCS	0b	**Receiver Overflow Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
16	RWCS	0b	**Unexpected Completion Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
15	RWCS	0b	**Completer Abort Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
14	RWCS	0b	**Completion Time Out Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
13	RWCS	0b	**Flow Control Protocol Error Status Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
12	RWCS	0b	**Poisoned TLP Received Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
11:5	RsvdP	00h	Preserved
4	RWCS	0b	**Data Link Protocol Error Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
3:1	RsvdP	000b	Preserved
0	RO	0b	**Training Error Severity:** Not supported

12.2.55 Offset 110h: ERRCOR_STS—PCI Express* Correctable Error Status

This register reports the error status of individual correctable error sources on a PCI Express* device. An individual error status bit set to 1 indicates that a particular error has occurred. Software can clear the error status by writing a 1 to the respective bit.

Table 89. Offset 110h: ERRCOR_STS—PCI Express* Correctable Error Status

Bits	Type	Reset	Description
31:13	RsvdZ	0	**Reserved Zero:** Software must write 0 to these bits.
12	RWCS	0b	**Replay Timer Time-out Status:** The Intel® 41210 Serial to Parallel PCI Bridge sets this bit when replay timer time-out occurs.
11:9	RsvdZ	000b	**Reserved Zero:** Software must write 0 to these bits.
8	RWCS	0b	**Replay Number Rollover Status:** The 41210 sets this bit when the replay number rolls over from 11 to 00.
7	RWCS	0b	**Bad DLLP Status:** The 41210 sets this bit on CRC errors on DLLP.
6	RWCS	0b	**Bad TLP Status:** The 41210 sets this bit on CRC errors on TLP.
5:1	RsvdZ	0 0000b	**Reserved Zero:** Software must write 0 to these bits.
0	RWCS	0b	**Receiver Error:** The 41210 sets this bit when the physical layer detects a receiver error.

12.2.56 Offset 114h: ERRCOR_MSK—PCI Express* Correctable Error Mask

This register controls the reporting of individual correctable errors via ERR_COR message. A masked error (respective bit set in mask register) is not reported to the host bridge by the 41210. There is a mask bit corresponding to every bit in the Correctable Error Status Register (Offset 110h: ERRCOR_STS—PCI Express* Correctable Error Status).

Table 90. Offset 114h: ERRCOR_MSK—PCI Express* Correctable Error Mask

Bits	Type	Reset	Description
31:13	RsvdP	0	Preserved
12	RWS	0b	**Replay Timer Time-out Mask:** 0 = Not masked 1 = Masked
11:9	RsvdP	000b	Preserved
8	RWS	0b	**Replay Number Rollover Mask:** 0 = Not masked 1 = Masked
7	RWS	0b	**Bad DLLP Mask:** 0 = Not masked 1 = Masked
6	RWS	0b	**Bad TLP Mask:** 0 = Not masked 1 = Masked
5:1	RsvdP	0 0000b	Preserved
0	RWS	0b	**Receiver Error Mask:** 0 = Not masked 1 = Masked

12.2.57 Offset 118h: ADVERR_CTL—Advanced Error Control and Capability Register

This register gives the status and control for ECRC checks and also the pointer to the first uncorrectable error that happened.

Table 91. Offset 118h: ADVERR_CTL—Advanced Error Control and Capability Register

Bits	Type	Reset	Description
31:9	RsvdP	0	Preserved
8	RO	0b	**ECRC Check Enable:** Not supported—hard-wired to 0
7	RO	0b	**ECRC Check Capable:** Not supported—hard-wired to 0
6	RO	0b	**ECRC Generation Enable:** Not supported—hard-wired to 0
5	RO	0b	**ECRC Generation Capable:** Not supported—hard-wired to 0
4:0	ROS	0 0000b	**The First Error Pointer:** This field identifies the bit position of the first error reported in the Uncorrectable Error Status Register ("Offset 104h: ERRUNC_STS—PCI Express* Uncorrectable Error Status Register" on page 105). This register re-arms itself (which does not change its current value) as soon as the error status bit indicated by the pointer is cleared by the software by writing a 1 to that status bit.

12.2.58 Offset 11C–12Bh: HDR_LOG—PCI Express* Transaction Header Log

This register is the transaction header log for PCI Express* errors.

Table 92. Offset 11C–12Bh: HDR_LOG—PCI Express* Transaction Header Log

Bits	Type	Reset	Description
127:0	ROS	0	**Header of the PCI Express* Packet in Error:** As soon as an error is logged in this register, it remains locked for further error-logging until the software clears the status bit that caused the header log (in other words, until the error pointer is re-armed for logging again).

12.2.59 Offset 12Ch: PCIXERRUNC_STS—Uncorrectable PCI-X Status Register

Table 93. Offset 12Ch: PCIXERRUNC_STS—Uncorrectable PCI-X Status Register (Sheet 1 of 2)

Bits	Type	Reset	Description
15:14	RsvdZ	00b	**Reserved Zero:** Software must write 0 to these bits.
13	RWCS	0b	**Internal Bridge Data Error:** This bit is set when an error occurs in the internal data queues in the Intel® 41210 Serial to Parallel PCI Bridge in either direction. The 41210 does ***not*** log any headers for this error.
12	RWCS	0b	**PCI-X SERR# Detected:** The 41210 sets this bit whenever it detects that the PCI SERR# pin is asserted. There is no header logging associated with the setting of this bit.
11	RWCS	0b	**PCI-X PERR# Detected:** The 41210 sets this bit whenever it detects that the PCI bus PERR# pin is asserted when it is mastering a write (memory, I/O, or configuration) or sourcing data during a split/delayed read completion on its secondary interface. The 41210 logs the header of the transaction in which the PERR# is detected (regardless of the data phase in which it is detected) in the PCI-X header log register ("Offset 11C–12Bh: HDR_LOG—PCI Express* Transaction Header Log" on page 110). This bit is also set when the bridge receives a PCI-X Split Completion Message with write data parity error status. The header log under this condition is the command, address, and attribute portion of the Split Completion Message. **NOTE:** This status bit and the associated header log are always updated regardless of whether the PERR# detected was the result of a PCI bus error or of forwarded poisoned data. However, error messages are not escalated to PCI Express* when the PERR# detection is due to forwarded poisoned data.
10	RWCS	0b	**PCI Delayed Transaction Timer Expiry:** This bit is set by the 41210 when it detects that a DT time-out has occurred on a delayed read stream or on an upstream I/O or configuration transaction. No header is logged.
9	RWCS	0b	**PCI-X Uncorrectable Address Parity Error Detected:** The 41210 sets this bit when it is the target of an upstream transaction and an address parity error is detected by the 41210 (regardless of whether the bus mode is PCI or PCI-X). The 41210 logs the header of the transaction in which it detected the address/attribute parity error in the PCI-X header log register.
8	RWCS	0b	**PCI-X Uncorrectable Attribute Parity Error Detected:** The 41210 sets this bit when it is the target of an upstream transaction and an attribute parity error is detected by the 41210. The 41210 logs the header of the transaction in which it detected the address/attribute parity error in the PCI-X header log register.
7	RWCS	0b	**PCI-X Uncorrectable Data Parity Error Detected:** The 41210 sets this bit in all PCI modes (PCI, PCI-X) when it is the target of a transaction or when it is mastering a PCI delayed read with target sourcing data to the 41210, and a data parity error was detected by the 41210. The 41210 logs the header of the transaction in which it detected the data parity error in the PCI-X header log register.
6	RWCS	0b	**Split Completion Message Data Error:** This bit is set when a split completion message is received with an uncorrectable data parity error.
5	RWCS	0b	**Unexpected Split Completion:** This bit is set when a completion is received from PCI-X that matches the bus number range on the primary side of the 41210 but the Requester ID:tag combination does not match one of the NP transmissions that the 41210 has outstanding on PCI-X.
4	RsvdZ	0b	**Reserved Zero:** Software must write a 0 to this bit.
3	RWCS	0b	**PCI-X Detected Master Abort:** The 41210 sets this bit when it is the master of a request transaction on the PCI-X bus and it receives a master abort. The 41210 logs the header for that transaction. This bit is also set when the bridge receives a PCI-X Split Completion Message with Master Abort Status. The header log under this condition is the command, address, and attribute portion of the Split Completion Message.

Table 93. Offset 12Ch: PCIXERRUNC_STS—Uncorrectable PCI-X Status Register (Sheet 2 of 2)

Bits	Type	Reset	Description
2	RWCS	0b	**PCI-X Detected Target Abort (optional in specification):** The 41210 sets this bit when it is the master of a request transaction on the PCI bus and it receives a target abort. The 41210 logs the header for that transaction. This bit is also set when the bridge receives a PCI-X Split Completion Message with Target Abort Status. The header log under this condition is the command, address, and attribute portion of the Split Completion Message.
1	RWCS	0b	**PCI-X Detected Split Completion Master Abort:** The 41210 sets this bit when a split completion sent by the 41210 on the PCI-X bus master-aborts. The 41210 logs the header of the split completion.
0	RWCS	0b	**PCI-X Detected Split Completion Target Abort (optional in specification):** The 41210 sets this bit when a split completion sent by the 41210 on the PCI-X bus target-aborts. The 41210 logs the header.

12.2.60 Offset 130h: PCIXERRUNC_MSK—Uncorrectable PCI-X Error Mask Register

This register masks the reporting of PCI-X uncorrectable errors. There is one mask bit per error. Note that the status bits are set in the status register regardless of whether the mask bit is on or off. The mask bit also affects the header log for the PCI-X transaction. When the mask bit is on, the header is not logged and no error message is generated on PCI Express*.

Table 94. Offset 130h: PCIXERRUNC_MSK—Uncorrectable PCI-X Error Mask Register (Sheet 1 of 2)

Bits	Type	Reset	Description
15:14	RsvdP	00b	Preserved
13	RWCS	0b	**Internal Bridge Data Error Mask:** 0 = Not masked 1 = Masked
12	RWCS	0b	**PCI-X SERR# Detected Mask:** 0 = Not masked 1 = Masked
11	RWCS	0b	**PCI-X PERR# Detected Mask:** 0 = Not masked 1 = Masked
10	RWCS	0b	**PCI Delayed Transaction Timer Expiry Mask:** 0 = Not masked 1 = Masked
9	RWCS	0b	**PCI-X Uncorrectable Address Parity Error Detected Mask:** 0 = Not masked 1 = Masked
8	RWCS	0b	**PCI-X Uncorrectable Attribute Parity Error Detected Mask:** 0 = Not masked 1 = Masked
7	RWCS	0b	**PCI-X Uncorrectable Data Parity Error Detected Mask:** 0 = Not masked 1 = Masked
6	RWCS	0b	**Split Completion Message Data Error Mask**
5	RWS	1b	**Unexpected Split Completion Error Mask**
4	RsvdP	0b	Preserved
3	RWCS	0b	**PCI-X Detected Master Abort Mask:** 0 = Not masked 1 = Masked

Table 94. Offset 130h: PCIXERRUNC_MSK—Uncorrectable PCI-X Error Mask Register (Sheet 2 of 2)

Bits	Type	Reset	Description
2	RWCS	0b	**PCI-X Detected Target Abort Mask: (optional in specification)** 0 = Not masked 1 = Masked
1	RWCS	0b	**PCI-X Detected Split Completion Master Abort Mask:** 0 = Not masked 1 = Masked
0	RWCS	0b	**PCI-X Detected Split Completion Target Abort Mask: (optional in specification)** 0 = Not masked 1 = Masked

12.2.61 Offset 134h: PCIXERRUNC_SEV—Uncorrectable PCI-X Error Severity Register

This register controls the severity of the reporting of PCI-X uncorrectable errors. There is one mask bit per error. When a bit is set to 1, the corresponding error, when enabled, generates an ERR_FATAL message on PCI Express*. When a bit is cleared to 0, the corresponding error, when enabled, causes a ERR_NONFATAL on PCI Express*.

Table 95. Offset 130h: PCIXERRUNC_SEV—Uncorrectable PCI-X Error Severity Register (Sheet 1 of 2)

Bits	Type	Reset	Description
15:14	RsvdP	00b	Preserved
13	RWCS	0b	**Internal Bridge Data Error Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
12	RWCS	0b	**PCI-X SERR# Detected Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
11	RWCS	0b	**PCI-X PERR# Detected Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
10	RWCS	0b	**PCI Delayed Transaction Timer Expiry Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
9	RWCS	0b	**PCI-X Uncorrectable Address Parity Error Detected Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
8	RWCS	0b	**PCI-X Uncorrectable Attribute Parity Error Detected Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
7	RWCS	0b	**PCI-X Uncorrectable Data Parity Error Detected Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
6	RWS	1b	**Split Completion Message Data Error Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
5	RWS	0b	**Unexpected Split Completion Error Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
4	RsvdP	0b	Preserved
3	RWCS	0b	**PCI-X Detected Master Abort Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL

Table 95. Offset 130h: PCIXERRUNC_SEV—Uncorrectable PCI-X Error Severity Register (Sheet 2 of 2)

Bits	Type	Reset	Description
2	RWCS	0b	**PCI-X Detected Target Abort Severity: (optional in specification)** 0 = ERR_NONFATAL 1 = ERR_FATAL
1	RWCS	0b	**PCI-X Detected Split Completion Master Abort Severity:** 0 = ERR_NONFATAL 1 = ERR_FATAL
0	RWCS	0b	**PCI-X Detected Split Completion Target Abort Severity: (optional in specification)** 0 = ERR_NONFATAL 1 = ERR_FATAL

12.2.62 Offset 138h: PCIXERRUNC_PTR—Uncorrectable PCI-X Error Pointer

This register points to the first error that occurred. This register is re-armed when the error status register corresponding to the error indicated by this register is cleared by the software writing a 1 to the bit.

Table 96. Offset 138h: PCIXERRUNC_PTR—Uncorrectable PCI-X Error Pointer Register

Bits	Type	Reset	Description
15:4	RsvdP	000h	Preserved
3:0	ROS	0h	**PCI First Error Pointer:** This register points to the first error that is logged in the status register (as long as the corresponding mask bit is clear and the pointer is re-armed). This register re-arms itself when the status bit corresponding to the error indicated by this register is cleared by the software writing a 1 to the bit.

12.2.63 Offset 13C–14Bh: PCIXHDR_LOG—Uncorrectable PCI-X Error Transaction Header Log

This register is the transaction header log for PCI errors. The log in this register corresponds to one of the status bits set in the PCI-X uncorrectable status register. As soon as an error is logged in this register, it remains locked for further error logging until the software clears the status bit that caused the header log (in other words, the error pointer is re-armed to log again).

Table 97. Offset 13C–14Bh: PCIXHDR_LOG—Uncorrectable PCI-X Header Log

Bits	Type	Reset	Description
127:64	ROS	0	**Address:** These bits capture the 64-bit address of the transaction in which an error was detected. In case of a 32-bit address, the upper address bits are all 0s. The address is logged on all error conditions.
63:44	ROS	0	Reserved
43:40	ROS	0000b	**DAC Bus Command:** This field captures the value of C/BE[3:0] during the second address phase of a DAC transaction.
39:36	ROS	0000b	**Bus Command:** This field captures the value of C/BE[3:0] during the first address phase of the transaction.
35:0	ROS	0	**Attribute:** This field carries the attribute of the transaction. The field is arranged as <CBE[3:0]#:AD[31:0]> during the attribute phase of the transaction. When the bus is in PCI mode, this register is all 0s.

12.2.64 Offset 16Ah: ARB_CNTRL—Internal Arbiter Control Register

Table 98. Offset 16Ah: ARB_CNTRL—Internal Arbiter Control Register

Bits	Type	Reset	Description
15:9	RsvdP	0	Preserved
8	RW	0b	**Bus Parking Control:** This bit controls the bus-parking behavior of the internal arbiter: 0 = The bus is parked on the last PCI agent using the bus. 1 = The bus is always parked on the Intel® 41210 Serial to Parallel PCI Bridge.
7	RW	1b	**Bridge Priority Ring Allocation:** This bit indicates the priority ring allocation for the 41210 bridge requests: 0 = The 41210 is in the low-priority ring. 1 = The 41210 is in the high-priority ring.
6:0	RW	7Fh	**PCI Master Priority Ring Allocation:** Bit[0] corresponds to REQ#[0], bit[1] corresponds to REQ#[1], and so on: 0 = The corresponding master is in the low-priority ring of the internal arbiter. 1 = The corresponding master is in the high-priority ring of the internal arbiter.

12.2.65 Offset 170h: SSR—Strap Status Register

This register indicates the status of various reset straps in the 41210.

Table 99. Offset 170h: SSR—Strap Status Register

<table>
<tr><th>Bits</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>15</td><td>RO</td><td>Strap</td><td>Configuration Retry Strap: This bit captures the CFGRETRY strap value at the rising edge of PERST#.</td></tr>
<tr><td>14:8</td><td>RO</td><td>00h</td><td>Reserved: Read only</td></tr>
<tr><td>7:1</td><td>RO</td><td>Strap</td><td>SMBus Address (SA): These seven bits represent the address to which the SMBus slave port responds when an access is attempted. This register has the following value:

<table><tr><th>Bit</th><th>Value</th></tr><tr><td>7</td><td>1</td></tr><tr><td>6</td><td>1</td></tr><tr><td>5</td><td>SMBUS[5]</td></tr><tr><td>4</td><td>0</td></tr><tr><td>3</td><td>SMBUS[3]</td></tr><tr><td>2</td><td>SMBUS[2]</td></tr><tr><td>1</td><td>SMBUS[1]</td></tr></table>
Only the value from function 0 is valid.</td></tr>
<tr><td>0</td><td>RO</td><td>Strap</td><td>P133EN Status: This bit reflects the status of the X_133EN pin sampled at the rising edge of PERST#.</td></tr>
</table>

12.2.66 Offset 178h: PREFCTRL—Prefetch Control Register

The following register contains prefetch parameters for PCI operation.

Table 100. Offset 178h: PREFCTRL—Prefetch Control Register

Bits	Type	Reset	Description
63:60	RsvdP	0000b	Preserved
55:54	RW	01b	**Prefetch Policy (PP):** These bits control how the bridge prefetches data on behalf of PCI masters: • 00 = Allow prefetching on MRM, MRL, and MR. • 01 = Allow prefetching on MRM and MRL, but not on memory read. • 1x = Disable all prefetching. **NOTE:** This control applies to both inbound memory reads and peer reads to the other PCI segment in the Intel® 41210 Serial to Parallel PCI Bridge.
53:22	RsvdP	0	Preserved
21:16	RW	05h	**TH 66:** These bits indicate the threshold parameter for 66 MHz PCI. Unit is 64B chunks and is 0s based; in other words, a value of 0 in this field indicates 64B.
15:11	RsvdP	00h	Preserved
10:5	RW	03h	**TH 33:** These bits indicate the threshold parameter for 33 MHz PCI. Unit is 64B chunks and is 0s based; in other words, a value of 0 in this field indicates 64B.
4:0	RsvdP	00h	Preserved

12.2.67 Offset 300h: PWRBGT_CAPID—Power Budgeting Enhanced Capability Header

This register defines the capability identifier.

Table 101. Offset 300h: PWRBGT_HDR—Power Budgeting Enhanced Capability Header

Bits	Type	Reset	Description
31:20	RO	000h	**Next PCI Express* Extended Capability Pointer:** This field indicates the last capability.
19:16	RO	1h	**Power Budgeting Capability Version Number:** This field indicates the PCI Express* Power Budgeting Capability Version Number.
15:0	RO	0004h	**Power Budgeting Capability ID:** This field indicates the Advanced Error Reporting Capability.

12.2.68 Offset 304h: PWRBGT_DSEL—Power Budgeting Data Select Register

Table 102. Offset 304h: PWRBGT_DSEL—Power Budgeting Data Select Register

Bits	Type	Reset	Description
7:0	RW	00h	**Data Select:** This read-write register indexes the reported through the Data Register ("Offset 308h: PWRBGT_DATA—Power Budgeting Data Register" on page 120) and selects the Dword of the power budgeting data that must appear in the Data Register. The index values for this register start at 0 to select the first DWord of the power budgeting data; subsequent DWords of the power budgeting data are selected by increasing index values. For example: • A value of 0 selects the DWord data starting at address 314h to appear in the Data Register at offset 308h. • A value of 1 selects the DWord data starting at address 318h to appear in the Data Register. • (and so on) When the Data Select field values is greater than 23, all zeros are reported in the Data Register.

12.2.69 Offset 308h: PWRBGT_DATA—Power Budgeting Data Register

Table 103. Offset 308h: PWRBGT_DATA—Power Budgeting Data Register

Bits	Type	Reset	Description
31:0	RO	0000 0000h	**Data:** This read-only register returns the DWord of the power budgeting data selected by the Data Select Register ("Offset 304h: PWRBGT_DSEL—Power Budgeting Data Select Register" on page 120). The Intel® 41210 Serial to Parallel PCI Bridge reports its power consumption for PM states D0, D3; types idle, sustained, maximum; power rails 1.2 V, 3.3 V, 1.8 V, Thermal.

§ §